

**KLASIFIKASI JENIS AUDIO BERDASARKAN KONDISI
PSIKOLOGI MENGGUNAKAN KOMBINASI *ALGORITME SELF
ORGANIZING MAPS* DAN *LEARNING VECTOR
QUANTIZATION***

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Rayhan Tsani Putra
NIM: 145150201111151



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

KLASIFIKASI JENIS AUDIO BERDASARKAN KONDISI PSIKOLOGI MENGGUNAKAN
KOMBINASI ALGORITME *SELF ORGANIZING MAPS* DAN *LEARNING VECTOR*
QUANTIZATION

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Rayhan Tsani Putra
NIM: 145150201111151

Skripsi ini telah diuji dan dinyatakan lulus pada
19 Januari 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Imam Cholissodin, S.Si, M.Kom

NIK: 201201 850719 1 001

Dosen Pembimbing II



Candra Dewi, S.Kom, M.Sc

NIP: 19771114 200312 2 001

Mengetahui
Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T, M.T., Ph.D

NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 19 Januari 2018



Rayhan Tsani Putra

NIM: 145150201111151

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT karena atas segala rahmat dan karunia Nya penulis dapat menyelesaikan laporan skripsi berjudul "Klasifikasi Jenis Audio Berdasarkan Kondisi Psikologi Menggunakan Kombinasi Algoritme *Self Organizing Maps* dan *Learning Vector Quantization*".

Dalam pengerjaan skripsi dan penyusunan laporan skripsi banyak pihak yang membantu, membimbing, serta memberikan dukungan baik secara moril maupun materiil. Oleh karena itu, penulis mengucapkan terimakasih yang sebesar-besarnya kepada:

1. Tri Astoto Kurniawan, S.T, M.T., Ph.D selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya Malang.
2. Agus Wahyu Widodo, S.T, M.Cs selaku Ketua Prodi Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya Malang.
3. Imam Cholissodin, S.Si, M.Kom selaku dosen pembimbing 1 skripsi yang telah membimbing dengan sabar, telaten dan sangat membantu dalam pengerjaan skripsi.
4. Candra Dewi, S.Kom, M.Sc selaku dosen pembimbing 2 skripsi yang telah membimbing dan memberi pengetahuan saat pengerjaan skripsi.
5. Ibu Effiana Yuriastien, S.Si, Psikolog selaku sumber informasi dan juga pakar.
6. Orang tua penulis, atas segala dukungan yang telah diberikan kepada kami.
7. Teman-teman dari Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya dan lainnya.
8. Danastri Ramya Mehaninda yang telah memberikan dukungan.
9. Seluruh pihak yang telah membantu kelancaran pelaksanaan dan penyusunan laporan skripsi ini.

Penulis menyadari bahwa dalam penyusunan laporan skripsi ini masih banyak kekurangan baik dari segi susunan maupun isi. Oleh karena itu, penulis dengan senang hati dan terbuka menerima kritik dan saran yang membangun sebagai pedoman perbaikan penulis.

Malang, 19 Januari 2018



Penulis

Rayhan Tsani Putra

Rayhantsani01@gmail.com

ABSTRAK

Karakteristik dari beberapa jenis audio memiliki pengaruh yang berbeda-beda terhadap emosi manusia dan juga aktifitas apa yang sedang dilakukan. Jika jenis audio atau musik yang didengarkan sesuai dengan kegiatan yang dilakukan dan juga dalam kondisi emosi yang sesuai dengan jenis kegiatan tersebut, maka sangat menguntungkan bagi pendengarnya. Biasanya masyarakat mendengarkan musik yang sudah biasa didengarkan tanpa memperdulikan kondisi yang tepat. Akan lebih baik jika dapat memanfaatkan dan memaksimalkan dampak positif dari audio tersebut dengan cara mendengarkan audio atau musik yang sesuai dengan kondisi. Pemilihan jenis audio secara otomatis berdasarkan *input* dari pengguna akan sangat mempermudah dan membantu seseorang dalam menentukan jenis audio yang sesuai. Klasifikasi adalah suatu cara pemilihan dari kelompok yang memiliki kemiripan pola dan menghasilkan suatu keluaran berupa nama label suatu kelas. Klasifikasi jenis audio akan sangat membantu dalam menentukan jenis audio yang sesuai. Penelitian ini mengklasifikasikan jenis audio berdasarkan salah satu kondisi psikologi yaitu emosi dan juga beberapa jenis aktifitas menggunakan kombinasi algoritme *SOM-LVQ (Self Organizing Map dan Learning Vector Quantization)*. *SOM* digunakan sebagai algoritme yang mendampingi dan melatih bobot awal untuk *LVQ* dikarenakan memiliki struktur dan alur kerja yang hampir sama dengan *LVQ*. Fitur yang digunakan pada penelitian ini sebanyak 11 yang terdiri dari kondisi psikologi dan jenis kegiatan. Terdapat 4 jenis audio yang menjadi kelas pada penelitian ini. Hasil evaluasi menunjukkan bahwa klasifikasi jenis audio menggunakan kombinasi *SOM-LVQ* lebih baik daripada hanya *LVQ* saja, namun data pada penelitian ini bersifat sensitif yang menyebabkan nilai akurasi kurang optimal. Nilai akurasi maksimal yang didapatkan pada penelitian ini adalah 89,583%. Kombinasi algoritme *SOM-LVQ* mencapai nilai akurasi maksimal tersebut dengan 4 iterasi pelatihan, sedangkan *LVQ* murni 6 iterasi maksimal. Walaupun dengan nilai akurasi yang sama, *SOM-LVQ* lebih baik dalam kecepatan mendapatkan nilai optimal (iterasi lebih kecil).

Kata Kunci: klasifikasi jenis audio, *Learning Vector Quantization*, *Self Organizing Map*, kondisi psikologi

ABSTRACT

The characteristics of each type of audio have different effects on human emotions as well as what activities are being performed. If the type of audio or music that is listened to in accordance with the activities undertaken and also in an emotional state appropriate to the type of activity, it is very beneficial for the listener. The most common case in most societies is listening to music that has been commonly heard without caring about the right conditions. It would be better if you can utilize and maximize the positive impact of the audio by listening to audio or music in accordance with the conditions. Automatic selection of audio types based on user input will greatly simplify and assist a person in determining the appropriate type of audio. Classification is a way of selecting from a group that has a similarity pattern and produces an output in the form of a class label name. Classification of audio types will be very helpful in determining the appropriate audio type. This study classifies the type of audio based on one of the psychological conditions of emotion and also some types of activities using a combination of SOM-LVQ algorithms (Self Organizing Map and Learning Vector Quantization). SOM is used as an algorithm that accompanies and trains initial weights for LVQ because it has a structure and workflow similar to LVQ. Feature used in this research is 11 which consist of psychology condition and activity type. There are 4 types of audio that became the class in this study. Evaluation results show that the classification of audio type using SOM-LVQ combination is better than only LVQ, but the data in this research is sensitive which causes less accurate accuracy value. The maximum accuracy obtained in this study was 89.583%. The SOM-LVQ algorithm combination achieves the maximum accuracy with 4 training iterations, while LVQ requires 6 iterations to achieve maximum value. Although with the same accuracy, SOM-LVQ is better at getting the optimal value.

Keywords: *classification of audio type, Learning Vector Quantization, Self Organizing Map, psychological condition*

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
<i>ABSTRACT</i>	vi
DAFTAR ISI	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Tujuan.....	3
1.4 Manfaat	3
1.5 Batasan Masalah	3
1.6 Sistematika Pembahasan	4
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Suara dan Audio	8
2.2.1 Jenis Audio.....	8
2.2.2 Musik	9
2.2.3 Murottal Al-Qur'an.....	10
2.3 Psikologi Manusia.....	10
2.4 Jaringan Syaraf Tiruan	12
2.5 Self Organizing Maps.....	13
2.6 Learning Vector Quantization	14
2.7 Kombinasi <i>SOM</i> dan <i>LVQ</i>	16
2.8 Nilai Evaluasi	17
BAB 3 METODOLOGI	18
3.1 Studi Pustaka	19
3.2 Pengumpulan Data.....	19
3.3 Analisis Kebutuhan	19

3.4 Perancangan.....	20
3.5 Implementasi.....	20
3.6 Pengujian.....	21
3.7 Kesimpulan dan Saran.....	21
BAB 4 PERANCANGAN.....	22
4.1 Deskripsi Masalah	22
4.2 Perancangan Klasifikasi	22
4.3 Perancangan Algoritme.....	23
4.3.1 <i>Self Organizing Maps</i>	23
4.3.2 <i>Learning Vector Quantization</i>	26
4.3.3 Contoh Manualisasi Pelatihan <i>SOM-LVQ</i>	28
4.4 Perancangan Uji Coba dan Evaluasi	39
4.4.1 Perancangan Uji Coba <i>SOM</i>	39
4.4.2 Perancangan Uji Coba <i>LVQ</i>	40
4.4.3 Perancangan Uji Coba <i>SOM-LVQ</i>	42
4.5 Perancangan Antarmuka.....	44
BAB 5 IMPLEMENTASI	45
5.1 Implementasi Data	45
5.2 Implementasi Algoritme.....	48
5.2.1 <i>Self Organizing Maps</i>	48
5.2.2 <i>Learning Vector Quantization</i>	52
5.3 Implementasi Antarmuka	56
BAB 6 PENGUJIAN DAN ANALISIS.....	57
6.1 Skenario Pengujian.....	57
6.2 Pengujian <i>SOM</i>	57
6.2.1 Pengujian <i>Learning Rate</i> Pada <i>SOM</i>	57
6.2.2 Pengujian <i>Decrement Learning Rate</i> Pada <i>SOM</i>	58
6.2.3 Pengujian Iterasi Pada <i>SOM</i>	60
6.3 Pengujian <i>LVQ</i>	61
6.3.1 Pengujian <i>Learning Rate</i> Pada <i>LVQ</i>	61
6.3.2 Pengujian <i>Decrement Learning Rate</i> Pada <i>LVQ</i>	63
6.3.3 Pengujian Iterasi Pada <i>LVQ</i>	64

6.4 Pengujian <i>SOM-LVQ</i>	65
6.4.1 Pengujian <i>Learning Rate</i> Pada <i>SOM-LVQ</i>	65
6.4.2 Pengujian <i>Decrement Learning Rate</i> Pada <i>SOM-LVQ</i>	67
6.4.3 Pengujian Iterasi Pada <i>SOM-LVQ</i>	68
6.5 Analisis dan Perbandingan	70
BAB 7 PENUTUP	72
7.1 Kesimpulan	72
7.2 Saran	72
DAFTAR PUSTAKA	73
LAMPIRAN A HASIL WAWANCARA	74
A.1 Wawancara Pakar	74
A.1 Wawancara Pengumpulan Data	75
LAMPIRAN B DATA	77
B.1 Data Awal (D1)	77
B.2 Data Awal (D2)	79
B.3 Data Lanjutan (D3)	84
B.4 Data Lanjutan (D4)	86

DAFTAR TABEL

Tabel 2.1 Perbedaan Karakteristik Penelitian	7
Tabel 4.1 Contoh Data Latih.....	29
Tabel 4.2 Hasil Perhitungan Jarak Iterasi 1 <i>SOM</i>	30
Tabel 4.3 Hasil Perbaikan Bobot Iterasi 1 <i>SOM</i>	31
Tabel 4.4 Hasil Perhitungan Jarak Iterasi 2 <i>SOM</i>	32
Tabel 4.5 Hasil Perbaikan Bobot Iterasi 2 <i>SOM</i>	33
Tabel 4.6 Hasil Perhitungan Jarak Iterasi 1 <i>SOM-LVQ</i>	34
Tabel 4.7 Hasil Perbaikan Bobot Iterasi 1 <i>SOM-LVQ</i>	35
Tabel 4.8 Hasil Perhitungan Jarak Iterasi 2 <i>SOM-LVQ</i>	36
Tabel 4.9 Hasil Perbaikan Bobot Iterasi 2 <i>SOM-LVQ</i>	37
Tabel 4.10 Tabel Perhitungan Akurasi <i>SOM-LVQ</i>	38
Tabel 4.11 Uji Coba <i>Learning Rate SOM</i>	39
Tabel 4.12 Uji Coba <i>Decrement SOM</i>	40
Tabel 4.13 Uji Coba Iterasi <i>SOM</i>	40
Tabel 4.14 Uji Coba <i>Learning Rate LVQ</i>	41
Tabel 4.15 Uji Coba <i>Decrement LVQ</i>	41
Tabel 4.16 Uji Coba Iterasi <i>LVQ</i>	42
Tabel 4.17 Uji Coba <i>Learning Rate SOM-LVQ</i>	42
Tabel 4.18 Uji Coba <i>Decrement SOM-LVQ</i>	43
Tabel 4.19 Uji Coba Iterasi <i>LVQ</i>	43
Tabel 5.1 Contoh Implementasi Data	45
Tabel 5.2 <i>Source Code Dataset</i>	45
Tabel 5.3 <i>Source Code Pelatihan SOM</i>	48
Tabel 5.4 <i>Source Code Euclidean Distance</i>	50
Tabel 5.5 <i>Source Code Minimum Distance</i>	51
Tabel 5.6 <i>Source Code Update Bobot SOM</i>	51
Tabel 5.7 <i>Source Code Pelatihan LVQ</i>	52
Tabel 5.8 <i>Source Code Update Bobot LVQ</i>	55
Tabel 5.9 <i>Source Code Test</i>	55
Tabel 6.1 Data Bobot Pengujian.....	57
Tabel 6.2 Pengujian <i>Learning Rate SOM</i>	58

Tabel 6.3 Pengujian <i>Decrement SOM</i>	59
Tabel 6.4 Pengujian Iterasi Pada <i>SOM</i>	60
Tabel 6.5 Pengujian <i>Learning Rate LVQ</i>	62
Tabel 6.6 Pengujian <i>Decrement LVQ</i>	63
Tabel 6.7 Pengujian Iterasi Pada <i>LVQ</i>	64
Tabel 6.8 Pengujian <i>Learning Rate SOM-LVQ</i>	66
Tabel 6.9 Pengujian <i>Decrement SOM-LVQ</i>	68
Tabel 6.10 Pengujian Iterasi <i>SOM-LVQ</i>	69
Tabel 6.11 Perbandingan Hasil Akhir	70

DAFTAR GAMBAR

Gambar 2.1 Struktur Dasar Jaringan Syaraf Tiruan.....	13
Gambar 3.1 Diagram Blok Metodologi Penelitian	18
Gambar 3.2 Model Perancangan Klasifikasi Genre Musik	20
Gambar 3.3 Diagram Blok Pengujian Akurasi Sistem.....	21
Gambar 4.1 Diagram Alir Klasifikasi	22
Gambar 4.2 Diagram Alir Pelatihan <i>SOM</i>	24
Gambar 4.3 Diagram Alir Perhitungan Jarak.....	24
Gambar 4.4 Diagram Alir Jarak Terdekat	25
Gambar 4.5 Diagram Alir <i>Update</i> Bobot	26
Gambar 4.6 Diagram Alir Pelatihan <i>LVQ</i>	27
Gambar 4.7 Diagram Alir Test Data	28
Gambar 4.8 Perancangan Antarmuka.....	44
Gambar 5.1 Implementasi Antarmuka.....	56
Gambar 6.1 Grafik Pengujian <i>Learning Rate SOM</i>	58
Gambar 6.2 Grafik Pengujian <i>Decrement SOM</i>	60
Gambar 6.3 Grafik Pengujian Iterasi <i>SOM</i>	61
Gambar 6.4 Grafik Pengujian <i>Learning Rate LVQ</i>	62
Gambar 6.5 Grafik Pengujian <i>Decrement LVQ</i>	64
Gambar 6.6 Grafik Pengujian Iterasi <i>LVQ</i>	65
Gambar 6.7 Grafik Pengujian <i>Learning Rate SOM-LVQ</i>	67
Gambar 6.8 Grafik Pengujian <i>Decrement SOM-LVQ</i>	68
Gambar 6.9 Grafik Pengujian Iterasi <i>SOM-LVQ</i>	69
Gambar 6.10 Grafik Perbandingan	71

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Audio memiliki berbagai macam jenis atau biasa disebut genre yang dibedakan variasinya atau dilabelkan berdasarkan cara permainan, variasi alat penghasil suara, budaya dan unsur lainnya yang membuat jenis audio menjadi beraneka ragam (Hilmi, 2015). Ada berbagai macam jenis audio di dunia ini yang masing-masing jenis tersebut memiliki karakteristik yang unik (Ifadah & Aimah, 2012). Karakteristik dari masing-masing jenis audio tersebut unik dikarenakan perbedaan frekuensi, unsur budaya dan unsur lainnya yang membuat variasi karakteristik tersebut sangat mudah dibedakan (Sakrie, 2015). Para ilmuwan telah meneliti bahwa sebagian musik dapat meningkatkan kemampuan intelektual dan emosional kita, bahkan musik juga bisa digunakan sebagai media pengobatan terapi (Nugroho, 2013). Pengaruh audio terhadap emosi atau perasaan manusia tersebut terus diteliti dan menghasilkan suatu kesimpulan yaitu berpengaruh kepada keefektifan aktifitas yang dikerjakan oleh pendengarnya, bahkan sebagai terapi untuk ketenangan (Supradewi, 2010). Pada penelitian yang lebih khusus, audio bahkan digunakan untuk terapi pada dunia medis (Handayani, et al., 2014). Audio tidak selalu berbentuk suatu musik atau instrumen. Dalam dunia medis, Murottal Al-Qur'an atau lantunan ayat suci Al-Qur'an dalam bentuk audio juga digunakan untuk pasien dan terbukti ampuh untuk mengurangi kecemasan pada proses melahirkan (Handayani, et al., 2014). Para penikmat lantunan audio dan musik merasa bahwa keunikan dari karakteristik masing-masing jenis audio dan musik tersebut dapat dimanfaatkan sesuai kondisi yang dialami oleh pendengarnya (Ifadah & Aimah, 2012).

Menurut penelitian yang dilakukan oleh Mills (1996), karakteristik dari masing-masing jenis audio tersebut memiliki pengaruh yang berbeda-beda terhadap emosi manusia dan juga aktifitas apa yang sedang dilakukan. Jika jenis audio atau musik yang didengarkan sesuai dengan kegiatan yang dilakukan dan juga dalam kondisi emosi yang sesuai dengan jenis kegiatan tersebut, maka sangat menguntungkan bagi pendengarnya (Mills, 1996). Namun Mills (1996) juga mengemukakan bahwa ada hubungan yang signifikan antara agresivitas seseorang dengan musik yang iramanya cepat. Kebisingan memiliki pengaruh yang negatif terhadap konsentrasi, produktivitas, kapasitas kerja dan resiko kecelakaan, walaupun pada level suara yang rendah (Sailer & Hassenzahl, 2000).

Penelitian yang dilakukan oleh Supradewi (2010) memaparkan bahwa terdapat pengaruh antara musik dan kinerja otak. Namun penelitian tersebut hanya berfokus pada pengaruh musik dengan kegiatan akademis dan solusi tidak dijelaskan secara langsung. Penelitian yang dilakukan oleh Nugroho (2013) juga menjelaskan bahwa musik dapat mempengaruhi kondisi psikologi seseorang. Namun penelitian tersebut hanya berfokus pada pengaruh musik dengan kondisi psikologi seseorang. Kesimpulan dari penelitian tersebut hanya terbatas pada pengaruh tanpa memberikan solusi secara langsung. Kasus yang sering terjadi

pada sebagian besar masyarakat adalah mendengarkan musik yang sudah biasa didengarkan tanpa memperdulikan kondisi yang tepat. Kasus tersebut berkaitan dengan beberapa penelitian yang sudah dipaparkan mengenai dampak negatif yang juga ditimbulkan dari jenis audio atau musik yang tidak sesuai dengan kondisi. Akan lebih baik jika dapat memanfaatkan dan memaksimalkan dampak positif dari audio tersebut dengan cara mendengarkan audio atau musik yang sesuai dengan kondisi (Nugroho, 2013). Pemilihan jenis audio secara otomatis berdasarkan *input* dari pengguna akan sangat mempermudah dan membantu seseorang dalam menentukan jenis audio yang sesuai. Klasifikasi adalah suatu cara pemilihan dari kelompok yang memiliki kemiripan pola dan menghasilkan suatu keluaran berupa nama label suatu kelas (Putri & Hartati, 2016). Klasifikasi jenis audio akan sangat membantu dalam menentukan jenis audio yang sesuai.

Penelitian tentang klasifikasi jenis audio atau musik berdasarkan frekuensi suara sudah banyak dilakukan. Salah satunya adalah penelitian yang dilakukan oleh Putri dan Hartati (2016). Namun pada penelitian yang diusulkan ini, jenis audio diklasifikasikan berdasarkan kondisi psikologi. Harapan dari penelitian ini yaitu dapat menghubungkan kondisi psikologi dengan jenis audio yang cocok didengarkan, mempermudah dan membantu seseorang dalam menentukan jenis audio yang sesuai kondisi dengan adanya klasifikasi jenis audio berdasarkan salah satu kondisi psikologi yaitu emosi. Salah satu algoritme untuk klasifikasi berdasarkan kemiripan pola dari suatu data adalah *Learning Vector Quantization (LVQ)*. *Learning Vector Quantization (LVQ)* adalah salah satu algoritme dari jaringan syaraf tiruan yang dapat mengklasifikasikan data berdasarkan pola atau data latih yang sudah ada. Kesederhanaan, fleksibilitas dan efisiensi merupakan beberapa alasan penggunaan algoritme *LVQ* pada penelitian ini (Putri & Hartati, 2016). Namun kelemahan dari algoritme *LVQ* adalah sangat bergantung atau sensitif terhadap pemilihan vektor acuan yang juga digunakan sebagai bobot awal jaringan dalam proses pelatihan *LVQ* (Putri & Hartati, 2016). Vektor acuan merupakan perwakilan dari masing-masing kelas yang merepresentasikan suatu kelas dalam jaringan *LVQ* (Putri & Hartati, 2016).

Penentuan vektor acuan yang paling mudah dan umum digunakan adalah dengan memilih langsung sejumlah vektor input sebagai perwakilan dari masing-masing kelas pada proses pelatihan. Inisialisasi vektor acuan (bobot awal) dengan cara ini sangat sensitif terhadap tingkat akurasi karena ketidaktepatan dalam pemilihannya dapat menghasilkan akurasi yang buruk (Putri & Hartati, 2016). Alternatif lain untuk penentuan vektor acuan adalah dengan proses kluster data pelatihan terlebih dahulu dengan *Self Organizing Map (SOM)* (Putri & Hartati, 2016). *SOM* digunakan sebagai algoritme yang mendampingi *LVQ* untuk kombinasi dikarenakan memiliki struktur dan alur kerja yang hampir sama dengan *LVQ* (Chen, et al., 2012). Kombinasi algoritme *SOM-LVQ* pada penelitian ini digunakan dengan harapan dapat menghasilkan akurasi yang lebih baik.

1.2 Rumusan Masalah

Pemaparan latar belakang tersebut dapat menjadi alasan dirumuskannya rumusan masalah yang mendasari penelitian ini, diantaranya adalah sebagai berikut:

1. Bagaimana menerapkan kombinasi algoritme *Self Organizing Maps (SOM)* dan *Learning Vector Quantization (LVQ)* dalam klasifikasi jenis audio berdasarkan kondisi psikologi dan jenis kegiatan?
2. Bagaimana tingkat akurasi yang dihasilkan oleh penelitian ini?

1.3 Tujuan

Berdasarkan rumusan masalah yang telah dipaparkan, maka tujuan penelitian ini dapat didefinisikan, diantaranya adalah sebagai berikut:

1. Menerapkan kombinasi algoritme *Self Organizing Maps (SOM)* dan *Learning Vector Quantization (LVQ)* dalam klasifikasi jenis audio berdasarkan kondisi psikologi dan jenis kegiatan.
2. Menghitung tingkat akurasi yang dihasilkan oleh penelitian.

1.4 Manfaat

Penelitian ini merupakan salah satu bentuk sumbangsih penulis dalam memajukan teknologi, khususnya dalam bidang komputasi cerdas. Penelitian ini menghasilkan suatu *prototype* yang diharapkan dapat dikembangkan menjadi sistem yang lebih bermanfaat bagi masyarakat. Penelitian ini juga menjadi persyaratan penulis untuk melangkah ke jenjang yang lebih tinggi.

1.5 Batasan Masalah

Agar penelitian lebih terfokus dan tidak meluas, maka batasan masalah penelitian ini adalah sebagai berikut:

1. Data yang digunakan dalam penelitian ini berdasarkan hasil wawancara secara langsung atau melalui media online kepada orang-orang sekitar seperti teman, kerabat dan pakar. Batasan daerah untuk pengumpulan data yaitu di negara Indonesia, tepatnya Kota Malang, Bekasi, Bogor dan Jakarta.
2. Total fitur yang dijadikan masukkan dalam sistem ini berjumlah 11 yang terdiri dari 2 jenis fitur yaitu kondisi psikologi dan jenis aktifitas. Kondisi psikologi yang dijadikan fitur pada penelitian ini berupa emosi dan dibatasi menjadi 8, diantaranya adalah konsentrasi, tenang, semangat, ceria, bebas, bertenaga, sendu, senang. Aktifitas yang dijadikan fitur yaitu kegiatan berfokus, berolahraga dan istirahat.
3. Genre atau jenis audio yang dijadikan hasil akhir dibatasi menjadi 4, diantaranya adalah klasik, *murottal Al-Qur'an*, rock dan pop.
4. Data uji yang digunakan sama dengan data latih dikarenakan terbatasnya variasi dan jumlah data pada *dataset* yang digunakan. Pengujian akurasi dilakukan dengan menggunakan nilai maksimum pada semua *dataset*. Hal tersebut juga dilakukan untuk memilih data latih dari *dataset* terbaik.

1.6 Sistematika Pembahasan

Penyusunan penelitian ini menggunakan kerangka pembahasan yang tersusun sebagai berikut:

BAB 1 PENDAHULUAN

Bab ini berisi latar belakang, rumusan masalah, tujuan, batasan masalah serta manfaat klasifikasi jenis audio menggunakan algoritme *Self Organizing Maps (SOM)* dan *Learning Vector Quantization (LVQ)*.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini digunakan untuk mendeskripsikan dasar teori yang akan digunakan dan mendukung penelitian klasifikasi jenis audio menggunakan algoritme *Self Organizing Maps (SOM)* dan *Learning Vector Quantization (LVQ)*. Tinjauan pustaka ini terdiri dari suara-audio, musik, *murottal*, psikologi manusia, jaringan syaraf tiruan, *SOM*, *LVQ*, kombinasi *SOM-LVQ* dan nilai evaluasi.

BAB 3 METODOLOGI

Metodologi penelitian menguraikan metode dan langkah kerja yang dilakukan untuk pengklasifikasian jenis audio. Metodologi penelitian terdiri dari studi literatur, pengumpulan data, analisis kebutuhan, perancangan klasifikasi, implementasi, pengujian, kesimpulan dan saran.

BAB 4 PERANCANGAN

Perancangan klasifikasi jenis audio menggunakan algoritme *Self Organizing Maps (SOM)* dan *Learning Vector Quantization (LVQ)*. Contoh perhitungan manual dan skenario pengujian juga dijelaskan pada tahap ini.

BAB 5 IMPLEMENTASI

Mengimplementasikan dan membahas klasifikasi jenis audio menggunakan algoritme *Self Organizing Maps (SOM)* dan *Learning Vector Quantization (LVQ)* berdasarkan kondisi psikologi terkait perasaan manusia.

BAB 6 PENGUJIAN DAN ANALISIS

Pengujian dilakukan sesuai perancangan, serta menganalisis hasil pengujian pada klasifikasi jenis audio menggunakan algoritme *Self Organizing Maps (SOM)* dan *Learning Vector Quantization (LVQ)* yang telah diimplementasikan.

BAB 7 PENUTUP

Bab ini memuat kesimpulan berdasarkan hasil penelitian klasifikasi jenis audio menggunakan algoritme *Self Organizing Maps (SOM)* dan *Learning Vector Quantization (LVQ)* yang telah dilakukan serta memberikan saran yang berguna untuk penelitian berikutnya.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Jenis audio dan musik terbukti dapat mempengaruhi kondisi psikologi manusia, khususnya emosi manusia (Nugroho, 2013). Sebagian jenis musik dapat meningkatkan kemampuan intelektual dan emosional kita, bahkan musik juga bisa digunakan sebagai media pengobatan terapi (Nugroho, 2013). Namun, musik juga memiliki dampak yang negatif, maka dari itu sebaiknya kita memaksimalkan dampak positif musik terhadap psikologi kita daripada dampak negatifnya (Nugroho, 2013). Penelitian yang dilakukan oleh Nugroho (2013) dengan topik hubungan musik dengan psikologi manusia meneliti tentang pengaruh musik terhadap psikologi manusia seperti halnya musik mampu mengendalikan emosi dan perasaan seseorang. Penelitian tersebut secara singkat memaparkan hubungan antara kondisi psikologi seseorang dengan musik beserta fakta dan sumber yang jelas. Namun penelitian tersebut belum memberikan penjelasan atau solusi secara langsung mengenai bagaimana cara memaksimalkan dampak positif musik terhadap psikologi kita daripada dampak negatifnya. Penelitian yang dirancang ini diharapkan dapat memberikan solusi mengenai cara memaksimalkan dampak positif dari musik terhadap psikologi manusia dengan cara klasifikasi dari beberapa jenis musik.

Penelitian mengenai pengaruh jenis musik juga dilakukan oleh Sari dan Grashinta (2015) dengan topik pengaruh jenis musik terhadap performa kognitif. Penelitian tersebut meneliti tentang pengaruh musik terhadap kemampuan ingatan anak usia 7 hingga 11 tahun yang nantinya diberikan serangkaian huruf dan diperdengarkan lagu yang berbeda. Dari hasil eksperimental tersebut di catat seberapa besar pengaruh jenis musik terhadap ingatan jangka pendek. Penelitian ini menghasilkan suatu kesimpulan bahwa musik berpengaruh terhadap ingatan jangka pendek dan faktor budaya juga mempengaruhi jenis musik yang mempengaruhi para siswa. Namun pada penelitian tersebut peneliti hanya membatasi pengaruh 2 jenis musik terhadap performa kognitif saja. Pada penelitian ini jenis audio atau musik yang digunakan adalah 4 jenis dan aktifitas yang berkaitan adalah 3 jenis aktifitas. Penelitian ini tidak secara langsung meneliti antara pengaruh musik dengan kondisi psikologi seseorang, tetapi penelitian ini dirancang untuk memberikan keluaran berupa jenis audio yang sesuai dengan kondisi psikologi seseorang tersebut. Pengenalan pola, pengelompokan data dan klasifikasi dapat digunakan untuk memberikan suatu keluaran berupa jenis audio tersebut dan diharapkan dapat mempermudah dalam memberikan suatu solusi.

Berbagai macam algoritme klasifikasi telah banyak diteliti dalam kasus-kasus yang lain. Kombinasi untuk memberikan hasil yang lebih optimal dalam klasifikasi juga sudah banyak diteliti. *Learning Vector Quantization (LVQ)* adalah salah satu algoritme klasifikasi dari struktur jaringan syaraf tiruan yang biasa di kombinasi dan di optimasi oleh algoritme *Self Organizing Map (SOM)* (Putri & Hartati,

2016). Salah satu penelitian dengan algoritme kombinasi *Self Organizing Map (SOM)* dan *Learning Vector Quantization (LVQ)* pernah diteliti oleh Chen dan kawan-kawan (2012). Penelitian tersebut menggunakan kombinasi algoritme *Self Organizing Map (SOM)* dan *Learning Vector Quantization (LVQ)* untuk pengambilan keputusan robot sepak bola. *SOM* digunakan untuk menyediakan bobot awal yang sudah dilatih dan diperbaiki untuk pelatihan selanjutnya oleh algoritma klasifikasi *LVQ*. Pengambilan keputusan dilakukan dengan *LVQ* berdasarkan pembelajaran dari kondisi terkini. Hasil dari keputusan tersebut adalah tindakan dari robot, yaitu menyerang, persiapan menyerang, persiapan bertahan dan bertahan. Penelitian tersebut menjadi referensi untuk penelitian ini, khususnya mengenai kombinasi algoritma yang dipakai. Penelitian ini juga menggunakan kombinasi algoritme *Self Organizing Map (SOM)* dan *Learning Vector Quantization (LVQ)*.

Penelitian tentang klasifikasi jenis audio atau musik berdasarkan frekuensi suara sudah banyak dilakukan. Salah satunya adalah penelitian yang dilakukan oleh Putri dan Hartati (2016). Penelitian tersebut juga menggunakan kombinasi algoritme yang sama dengan penelitian ini yaitu *Self Organizing Map (SOM)* dan *Learning Vector Quantization (LVQ)* untuk klasifikasi. Objek pada penelitian tersebut berupa audio atau musik yang hasil keluarannya berupa genre musik. *SOM* digunakan untuk menyediakan bobot awal pada pelatihan *LVQ* dan *LVQ* digunakan untuk mengklasifikasikan jenis musik. Genre musik pada penelitian klasifikasi tersebut dibatasi menjadi 10 genre. Lalu fitur berdasarkan konten musik dibatasi menjadi 3 yaitu *timbral texture*, *rythm* dan *pitch*. Namun penelitian tersebut masih memberikan hasil persentase keakuratan yang kecil dikarenakan ketidaksesuaian fitur. Pada penelitian yang diusulkan ini, jenis audio diklasifikasikan berdasarkan kondisi psikologi. Harapan dari penelitian ini yaitu dapat menghubungkan kondisi psikologi dengan jenis audio yang cocok didengarkan, mempermudah dan membantu seseorang dalam menentukan jenis audio yang sesuai kondisi dengan adanya klasifikasi jenis audio berdasarkan salah satu kondisi psikologi yaitu emosi.

Beberapa penelitian terdahulu dijadikan referensi untuk penelitian yang dikerjakan saat ini. Unsur dari bahan yang dijadikan referensi dari beberapa penelitian sebelumnya untuk penelitian ini berupa topik yang serupa dengan penelitian, algoritme yang menunjang penelitian, alur kerja kombinasi algoritme yang menunjang penelitian, konsep dan juga objek serupa yang dijadikan penelitian. Pemilihan algoritme *SOM-LVQ* pada penelitian ini karena pada penelitian terdahulu algoritme *SOM-LVQ* berhasil diterapkan untuk klasifikasi genre musik, namun dengan fitur yang berbeda. Penelitian terdahulu memiliki kekurangan dan kelebihan tersendiri yang dijadikan acuan dan juga celah untuk diperbaiki pada penelitian ini. Penelitian ini diharapkan dapat memberikan beberapa solusi yang belum ada pada penelitian sebelumnya. Agar mempermudah dalam mengetahui dan membedakan unsur dari masing-masing penelitian, disusun sebuah tabel ringkasan yang berisi karakteristik dari penelitian-penelitian tersebut. Tabel ringkasan perbedaan karakteristik dapat dijabarkan pada Tabel 2.1.

Tabel 2.1 Perbedaan Karakteristik Penelitian

No	Referensi	Metode	Fitur	Hasil Keluaran
1	(Nugroho, 2013)	Penelitian psikologi dan pengambilan kesimpulan	Sumber referensi	Pengaruh dan manfaat musik terhadap psikologi manusia serta saran-saran tentang bagaimana memaksimalkan manfaat positif dari musik terhadap psikologi
2	(Sari & Grashinta, 2015)	Pencocokan data	Susunan huruf	Nilai kecocokan dari hasil tes dengan jawaban, jenis musik yang mempengaruhi dan juga rekomendasi jenis musik untuk anak sesuai budaya dan lingkungan.
3	(Chen, et al., 2012)	<i>Self Organizing Maps (SOM)</i> dan <i>Learning Vector Quantization (LVQ)</i>	Kondisi terkini	Hasil dari keputusan robot sepak bola yang berupa tindakan dari robot, yaitu menyerang, persiapan menyerang, persiapan bertahan dan bertahan.
4	(Putri & Hartati, 2016)	<i>Learning Vector Quantization (LVQ)</i> dan <i>Self Organizing Map (SOM)</i>	Timbral texture, rythm dan pitch	Hasil klasifikasi berupa genre musik, penentuan fitur terbaik dan kesimpulan bahwa akurasi yang lebih baik jika menggunakan metode gabungan LVQ dan SOM.
5	Usulan Penelitian	<i>Self Organizing Maps (SOM)</i> dan <i>Learning Vector Quantization (LVQ)</i>	Emosi, perasaan dan aktifitas yang sedang di alami atau ingin dicapai	Rekomendasi jenis audio, <i>link playlist</i> sesuai jenis audio, akurasi sistem berdasarkan data latih yang ada dan nilai fitur terbaik.

2.2 Suara dan Audio

Suara adalah pemampatan gelombang *longitudinal* yang merambat melalui medium yang berupa zat cair, gas, maupun padat (Hilmi, 2015). Umumnya suara merupakan gabungan beberapa sinyal, tetapi suara murni dapat dijelaskan dengan frekuensi yang diukur dalam satuan *Hertz (Hz)* dan *amplitudo* atau kenyaringan bunyi dengan pengukuran dalam desibel. Suara dihasilkan oleh getaran suatu benda. Selama benda bergetar, akan terjadi perbedaan tekanan udara pada lingkungan sekitar. Pola getaran yang terjadi disebut gelombang. Gelombang memiliki pola yang sama pada interval tertentu yang disebut periode (Hilmi, 2015).

Suara berdasarkan periodenya dibedakan menjadi dua, yaitu suara periodik dan suara non-periodik. Suara periodik terjadi secara berulang, seperti instrumen musik, kicauan burung dan lain sebagainya. Sedangkan suara non-periodik adalah suara yang tidak terjadi secara berulang dan hanya berbunyi satu atau beberapa kali, seperti bunyi batuk, benda jatuh dan lain sebagainya (Hilmi, 2015). Suara dibagi menjadi dua berdasarkan medianya, yaitu suara analog dan suara digital. Suara analog merupakan suara asli dari benda yang bergetar ataupun suara dari alam yang merambat melalui gelombang yang menjadi frekuensi (jumlah gelombang yang terjadi dalam satu detik). Sedangkan suara digital adalah suara yang dihasilkan melalui media komputer atau elektronik yang sudah terintegrasi dengan perangkat pembangkit sinyal dan suara. Sinyal suara pada perangkat komputer merupakan hasil representasi suara analog atau suara yang sesungguhnya melalui rekaman lalu disimpan dalam kode digital komputer (Hilmi, 2015).

Audio adalah sinyal elektrik yang digunakan untuk membawa unsur suara. Istilah ini juga bisa digunakan untuk menerangkan sistem-sistem yang berkaitan dengan proses perekaman dan transmisi, yaitu sistem pengambilan atau penangkapan suara. Audio merupakan representasi dari suara analog (suara yang sebenarnya) menjadi suara digital di dalam komputer melalui rekaman yang disimpan dalam kode digital di dalam bahasa komputer dengan format yang ada (Hilmi, 2015).

2.2.1 Jenis Audio

Suara digital yang juga bisa disebut sebagai audio dibagi menjadi beberapa tipe. Berdasarkan macamnya audio dibagi menjadi 5, antara lain (Hilmi, 2015):

1. Audio Visual
Perangkat *sound system* yang dilengkapi dengan penampil gambar, biasanya digunakan untuk presentasi, home theater dan lain sebagainya.
2. Audio Streaming
Suatu istilah yang digunakan untuk mendengarkan suatu siaran secara langsung melalui internet. Berbeda dengan cara lain yang mengharuskan mengunduh *file* lalu baru dijalankan, dengan cara *streaming* ini kita bisa mendengarkan langsung tanpa perlu mengunduh terlebih dahulu suatu *file* tersebut.

3. Audio Response
Output pembicaraan yang dihasilkan komputer untuk menanggapi *input* jenis khusus, misalnya permintaan nomor telepon.
4. Audio Oscillator
Audio pembuatan film yang digunakan pertama kali oleh perusahaan Walt Disney dalam pembuatan film Fantasia.
5. Audio Modem Riser
Sebuah kartu *plug-in* untuk *motherboard* komputer yang digunakan oleh perusahaan Intel. Memuat sirkuit audio dan sirkuit modem. Audio Modem Riser memuat fungsi-fungsi analog berupa kode-kode yang diperlukan untuk operasi modem dan audio.

Macam audio yang digunakan pada penelitian ini adalah suara digital atau audio *streaming*. Fokus audio streaming pada penelitian ini adalah kombinasi audio yang bisa berupa musik *instrumental*, musik berlibir dan lantunan ayat suci atau bacaan kitab.

2.2.2 Musik

Musik adalah penghayatan isi hati manusia yang diungkapkan menjadi bunyi yang teratur dengan ritme dan keselarasan yang indah (Widhyatama, 2012). Biasanya musik memang tampil berupa rangkaian nada, baik dalam vokal, instrumental dari alat-alat musik ataupun kombinasi keduanya. Lagu merupakan sebuah teks yang dinyanyikan (Ifadah & Aimah, 2012). Lagu berasal dari sebuah karya tulis yang diperdengarkan dengan iringan musik (Ifadah & Aimah, 2012). Namun musik dapat pula dihasilkan hanya dengan tepuk tangan, pukulan, decitan dan gesekan dari benda benda sekitar. Dalam hal ini semua itu hanyalah alat musik tak bernada, meskipun ada bunyinya. Dalam kaitan dengan wawasan seni, semua sumber seni, tidak semua sumber seni yang indah dapat dianggap sebagai karya seni, sebab yang disebut seni adalah sesuatu olah pikir, akal budi, curahan hati dan perasaan manusia (Widhyatama, 2012).

2.2.2.1 Genre Musik

Musik memiliki berbagai macam karakteristik yang membedakannya satu dengan lainnya. Pengelompokan musik sesuai dengan kemiripan karakteristiknya disebut sebagai genre musik. Karakteristik genre musik berbeda-beda bisa karena letak geografi, bahasa suatu daerah, budaya suatu daerah, perkembangan zaman, keragaman alat musik yang digunakan, teknik pembawaan, tema dan karakteristik lainnya (Widhyatama, 2012).

Seiring perkembangan zaman, genre musik pun mulai banyak macamnya dan bahkan ada *subgenre* yang merupakan kombinasi dari genre utama. Genre yang menjadi dasar dari *subgenre* tetaplah eksis dan bahkan semakin berkembang. Genre-genre tersebut diantaranya (Sakrie, 2015):

1. Klasik
Merupakan musik yang tumbuh dan berkembang dari daerah barat dan berasal dari tradisi kesenian barat yaitu musik sekuler dan orkestra. Musik

klasik awalnya juga tumbuh dari tradisi keagamaan masyarakat barat. Karakteristik dari musik klasik adalah musik yang terfokus pada tinggi nada, kecepatan, *metrum*, ritme, keserasian dan pembawaan yang tepat. Musik klasik biasanya berupa instrumental, kombinasi dari berbagai alat musik dan vokal bernada tinggi.

2. Rock

Musik rock adalah genre musik populer yang mulai eksis dan diketahui khalayak luas pada pertengahan tahun 1950-an. Musik klasik juga menjadi pencetus dan akar dari musik rock. Bunyi khas dari musik rock adalah nada *back beat* yang sangat dominan dari gitar listrik ataupun gitar akustik yang diikuti oleh dentuman drum dan bass, juga iringan dari piano terkadang mengiringi beberapa lagu rock.

3. Pop

Musik pop berasal dari kata populer yang mulai dikenal pada tahun 1950-an. Musik pop merupakan genre yang lebih lembut dari musik-musik yang bermunculan pada tahun 1940-an. Kekuatan dari musik ini adalah liriknya yang mudah dipahami, nada yang sangat bisa diterima oleh masyarakat umum dan juga disebut-sebut sebagai genre musik yang memiliki daya tarik penikmat dalam jumlah banyak. Terkadang pop memiliki nada yang semangat dan juga memiliki nada yang sedih. Pop sangat diterima oleh masyarakat hingga masa kini.

2.2.3 Murottal Al-Qur'an

Murottal adalah bacaan atau rekaman suara Al-Qur'an yang dilagukan oleh seorang *qori'*. Suara Al-Qur'an ibarat gelombang suara yang memiliki ketukan dan gelombang tertentu, menyebar dalam tubuh kemudian menjadi getaran yang bisa mempengaruhi fungsi gerak sel otak dan membuat keseimbangan didalamnya. Sesuatu yang terpengaruh dengan tilawah Al-Qur'an, getaran *neuron*-nya akan stabil kembali. Al-Qur'an mempunyai beberapa manfaat karena terkandung beberapa aspek yang dapat berpengaruh terhadap kesehatan antara lain: Mengandung unsur meditasi, autosugesti dan relaksasi (Ernawati, 2013).

Ada banyak ayat-ayat dalam Al-Qur'an yang menjelaskan tentang bagaimana cara serta pengaruh yang ditimbulkan saat seseorang mendengarkan bacaan Al-Qur'an baik pada seseorang yang mengerti makna dari bacaan tersebut ataupun tidak mengerti sama sekali. Mendengarkan bacaan Al-Qur'an akan berpengaruh jika didengarkan dalam keadaan yang tenang serta pendengar memperhatikan dalam arti tidak berbicara atau meninggalkan kesibukan yang dapat mengganggu dari mendengarkan. Selain itu pendengar juga harus menghadirkan hati untuk meresapi apa yang didengar (Ernawati, 2013).

2.3 Psikologi Manusia

Psikologi adalah bidang yang berkaitan erat dengan emosi, perasaan dan kejiwaan manusia (Yuriastien, 2017). Perasaan adalah bagian dari emosi manusia (Yuriastien, 2017). Emosi dan Perasaan manusia adalah salah satu pendekatan

untuk mengetahui dan mengatasi kondisi psikologi manusia (Yuriastien, 2017). Logika juga berkaitan dengan emosi dan perasaan manusia karena logika yang memiliki keterkaitan dalam proses yang nantinya menimbulkan efek perasaan dan emosi tersebut (Nagara, 2016). Menurut (Nagara, 2016) emosi dan perasaan sangat berperan penting dalam kehidupan karena mempengaruhi mental manusia tersebut. Mental pada manusia sangatlah penting karena berhubungan dengan kejiwaan yang menjadi inti dari kemanusiaan. Kejiwaan manusia sangatlah berpengaruh kepada aktifitas yang dilakukan setiap harinya didalam kehidupan.

Macam-macam emosi dan perasaan manusia sangatlah banyak, ada yang dapat di deskripsikan dengan penalaran ada juga yang tidak dapat (Yuriastien, 2017). Terkadang manusia ingin perasaan dan emosinya sesuai dengan kondisi atau keadaan sekitar agar membantunya untuk melakukan aktivitas (Nagara, 2016). Mereka yang mendengarkan musik bisa mengalami kondisi emosi seperti senang, sedih, semangat, gundah dan sebagainya (Ifadah & Aimah, 2012). Kondisi psikologi berupa emosi yang dijadikan fitur pada penelitian ini adalah hasil dari wawancara dengan psikolog (Yuriastien, 2017), penelitian mengenai pengaruh musik dan psikologi (Nugroho, 2013) dan juga penelitian mengenai pengaruh musik terhadap kognitif (Ifadah & Aimah, 2012). Beberapa kondisi psikologi tersebut antara lain:

1. Konsentrasi
Konsentrasi adalah perasaan yang timbul ketika seseorang sedang berfokus terhadap suatu hal yang sedang kita lakukan.
2. Tenang
Tenang adalah perasaan dimana seseorang merasa aman, nyaman, rileks dan sejenisnya.
3. Semangat
Semangat adalah perasaan dimana seseorang merasa harus mencapai sesuatu dan percaya dapat melakukannya.
4. Ceria
Ceria adalah perasaan dimana seseorang merasa berseri-seri, girang, gembira dan sejenisnya.
5. Bebas
Bebas adalah perasaan dimana seseorang merasa terlepas dari kerumunan, merasa lega dari suatu keramaian dan tidak ada gangguan.
6. Bertenaga
Bertenaga adalah perasaan dimana seseorang merasa menggebu-gebu, penuh tenaga untuk melakukan segala aktifitas.
7. Sendu
Sendu adalah perasaan dimana seseorang merasa sunyi, sedih, galau, gundah. Tetapi sendu juga bisa berarti khuyu saat sedang melakukan sesuatu.
8. Senang
senang adalah perasaan dimana seseorang merasa bahagia, suka, gemar dan perasaan sejenisnya.

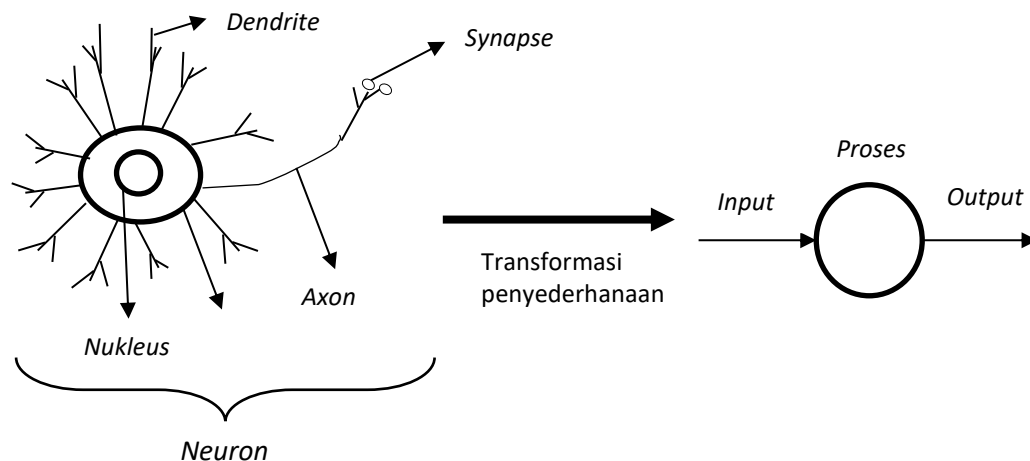
Emosi dan perasaan dapat mempengaruhi aktifitas yang sedang dilakukan oleh manusia, oleh karena itu emosi dan perasaan yang tepat dalam mengerjakan suatu aktifitas sangatlah penting (Mills, 1996). Aktifitas yang dijadikan fitur dalam penelitian ini ditentukan berdasarkan penelitian sebelumnya mengenai pengaruh musik terhadap performa fisik (Mills, 1996) dan pengaruh musik terhadap proses belajar (Supradewi, 2010). Beberapa aktifitas tersebut antara lain:

1. Berfokus
Berfokus adalah aktifitas dimana aktifitas tersebut membutuhkan konsentrasi, teliti dan fokus yang tinggi. Contoh aktifitas berfokus adalah mengerjakan suatu pekerjaan atau tugas, belajar, meneliti dan hal sejenisnya.
2. Berolahraga
Berolahraga adalah aktifitas dalam lingkup olahraga atau kebugaran fisik seperti bermain bola, lari pagi, *workout* dan hal sejenisnya.
3. Istirahat
Istirahat adalah aktifitas relaksasi yang bertujuan untuk mengistirahatkan tubuh dan pikiran. Contoh istirahat adalah tidur, duduk sejenak, berdoa, beribadah dan sejenisnya.

2.4 Jaringan Syaraf Tiruan

Suatu jaringan saraf tiruan memproses sejumlah besar informasi secara paralel dan terdistribusi, hal ini terinspirasi oleh model kerja otak biologis. Sistem Syaraf buatan adalah suatu struktur pemroses informasi yang terdistribusi dan bekerja secara paralel, yang terdiri atas elemen pemroses (yang memiliki memori lokal dan beroperasi dengan informasi lokal) yang diinterkoneksi bersama dengan alur sinyal searah yang disebut koneksi. Setiap elemen pemroses memiliki koneksi keluaran tunggal yang bercabang (*fan out*) ke sejumlah koneksi *kolateral* yang diinginkan (setiap koneksi membawa sinyal yang sama dari keluaran elemen pemroses tersebut).

Seluruh proses yang berlangsung pada setiap elemen pemroses harus benar-benar dilakukan secara lokal, yaitu keluaran hanya bergantung pada nilai masukan pada saat itu yang diperoleh melalui koneksi dan nilai yang tersimpan dalam memori lokal. Struktur pada Gambar 2.1 adalah bentuk standar dasar satuan unit jaringan otak manusia yang telah disederhanakan. Jaringan otak manusia tersusun dari 1013 *neuron* yang terhubung oleh sekitar 1015 *dendrite*. Fungsi *dendrite* adalah sebagai penyampai sinyal dari *neuron* tersebut ke *neuron* yang terhubung dengannya. *Nucleus* merupakan inti dari suatu *neuron*, *dendrite* merupakan tempat masuknya sinyal, *soma* berfungsi memproses sinyal yang masuk, *axon* berfungsi sebagai saluran keluaran dari *neuron*, dan *synapse* yang mengatur kekuatan hubungan antar *neuron*.



Gambar 2.1 Struktur Dasar Jaringan Syaraf Tiruan

Sumber: (Pakaja, et al., 2012)

Jaringan *neuron* buatan terdiri atas kumpulan grup *neuron* yang tersusun dalam lapisan. Lapisan *input* (*Input Layer*) berfungsi sebagai penghubung jaringan ke dunia luar (sumber data). Lapisan tersembunyi (*Hidden Layer*) suatu jaringan dapat memiliki lebih dari satu *hidden layer* atau bahkan bisa juga tidak memilikinya sama sekali. Lapisan keluaran (*Output Layer*) prinsip kerja *neuron-neuron* pada lapisan ini sama dengan prinsip kerja *neuron-neuron* pada lapisan tersembunyi (*hidden layer*) dan di sini juga digunakan fungsi *Sigmoid*, tapi keluaran dari *neuron* pada lapisan ini sudah dianggap sebagai hasil dari proses. Secara umum, terdapat tiga jenis *neural network* yang sering digunakan berdasarkan jenis *network*-nya, yaitu (Pakaja, et al., 2012):

1. *Single-Layer Neural* adalah jaringan syaraf tiruan yang memiliki koneksi pada masukan secara langsung ke jaringan *output*.
2. *Multilayer Perceptron Neural Network* adalah jaringan syaraf tiruan yang mempunyai layer yang dinamakan *hidden*, ditengah layer *input* dan *output*. *Hidden* ini bersifat variabel, dapat digunakan lebih dari satu *hidden layer*.
3. *Recurrent Neural Networks Neural network* adalah jaringan syaraf tiruan yang memiliki ciri, yaitu adanya koneksi umpan balik dari *output* ke *input*.

2.5 Self Organizing Maps

Self Organizing Maps (SOM) diperkenalkan oleh Teuvo Kohonen seorang ilmuwan Finlandia pada tahun 1982, sehingga *Self Organizing Maps (SOM)* dikenal juga dengan jaringan *Kohonen*. *Self Organizing Maps (SOM)* merupakan salah satu jaringan syaraf tiruan yang dikonfigurasi untuk pengelompokan data. Jaringan *SOM* atau *Kohonen* menggunakan metode pembelajaran *unsupervised* yang proses pelatihannya tidak memerlukan pengawasan (*target output*). Jaringan kohonen *SOM* terdiri dari 2 lapisan atau *layer*, yaitu lapisan *input* dan lapisan *output*. Setiap *neuron* dalam lapisan *input* terhubung dengan setiap

neuron pada lapisan *output*. Setiap *neuron* dalam lapisan *output* merepresentasikan kelompok dari *input* yang diberikan (Putri & Hartati, 2016).

Penulisan istilah yang ada pada struktur jaringan *kohonen Self Organizing Maps* adalah sebagai berikut (Azmi, 2014):

- i : Dimensi/kolom (fitur ke- i)
- j : Indeks/baris (data ke- j)
- α : Laju pembelajaran
- Dec : Nilai pengurang α
- W : Bobot pada *neuron*
- X : Vektor *input* pembelajaran
- W_{ji} : Indeks bobot ke- j pada dimensi ke- i
- X_{ji} : Indeks *input* ke- j pada dimensi ke- i
- D : Jarak antara bobot dan *input*
- D_j : Jarak antara bobot dan *input* pada indeks bobot ke- j

Langkah-langkah algoritme *SOM* melalui beberapa tahapan yang terurut, terdiri atas (Azmi, 2014):

1. Melakukan inisialisasi berupa:
 - Inisialisasi bobot W_{ji} .
 - Set nilai pengurang laju pembelajaran Dec .
 - Set parameter laju pembelajaran α .
 - Set maksimum iterasi (syarat berhenti)
2. Jika syarat berhenti tidak dipenuhi (Salah), Kerjakan langkah 3 – 9.
3. Untuk setiap *input* vektor X , kerjakan langkah 4 – 6.
4. Menghitung jarak setiap X dengan setiap W menggunakan rumus *Euclidean*, yaitu menggunakan Persamaan (2.1):

$$D_j = \sqrt{\sum_{i=1}^n (X_{ji} - W_{ji})^2} \quad (2.1)$$

5. Cari unit pemenang (indeks j), yaitu unit yang memiliki D_j minimum.
6. Perbaiki nilai W_{ji} sesuai dengan indeks j dari langkah 5 menggunakan Persamaan (2.2):

$$W_{ji}(\text{baru}) = W_{ji}(\text{lama}) + \alpha(X_{ji} - W_{ji}(\text{lama})) \quad (2.2)$$

7. Ubah (*update*) nilai laju pembelajaran menggunakan Persamaan (2.3):

$$\alpha(\text{baru}) = Dec * \alpha(\text{lama}) \quad (2.3)$$

8. Periksa syarat berhenti (iterasi).

2.6 Learning Vector Quantization

Learning Vector Quantization (LVQ) adalah salah satu algoritme pembelajaran dari jaringan syaraf tiruan untuk melakukan pembelajaran pada lapisan kompetitif yang terawasi. Suatu lapisan kompetitif akan secara otomatis belajar untuk mengklasifikasikan vektor-vektor *input*. Kelas-kelas yang didapatkan sebagai hasil dari lapisan kompetitif ini hanya tergantung pada jarak antara

vektor-vektor *input*. Jika dua vektor *input* mendekati sama, maka lapisan kompetitif akan meletakkan kedua vektor *input* tersebut kedalam kelas yang sama (Budianita & Arni, 2015).

Learning Vector Quantization (LVQ) adalah suatu metode klasifikasi pola masing-masing unit keluaran mewakili kategori atau kelas tertentu (beberapa unit keluaran seharusnya digunakan untuk masing-masing kelas). Vektor bobot untuk suatu unit keluaran sering dinyatakan sebagai sebuah vector *referens*. Diasumsikan bahwa serangkaian pola pelatihan dengan klasifikasi yang tersedia bersama dengan distribusi awal vektor *referens*. Setelah pelatihan, jaringan LVQ mengklasifikasi vektor masukan dengan menugaskan ke kelas yang sama sebagai unit keluaran, sedangkan yang mempunyai vektor *referens* diklasifikasikan sebagai vektor masukan (Budianita & Arni, 2015).

Penulisan istilah yang ada pada struktur jaringan *Learning Vector Quantization* adalah sebagai berikut (Azmi, 2014):

- i : Dimensi/kolom (fitur ke- i)
- j : Indeks/baris (data ke- j)
- α : Laju pembelajaran
- Dec : Nilai pengurang α
- W : Bobot pada *neuron*
- X : Vektor *input* pembelajaran
- W_{ji} : Indeks bobot ke- j pada dimensi ke- i
- X_{ji} : Indeks *input* ke- j pada dimensi ke- i
- D : Jarak antara bobot dan *input*
- D_j : Jarak antara bobot dan *input* pada indeks bobot ke- j
- C_j : Target dari bobot indeks ke- j

Proses pengklasifikasian menggunakan metode LVQ melalui beberapa tahapan yang terurut, terdiri atas (Azmi, 2014):

1. Menentukan *input*
2. Menentukan bobot awal, batas iterasi (*epoch*) dan α (*learning rate*)
3. Melakukan perhitungan apabila iterasi $\leq Max$ iterasi dan $\alpha \geq Min\alpha$
4. Menghitung jarak setiap X dengan setiap W menggunakan rumus *Euclidean*, yaitu menggunakan Persamaan (2.4):

$$D_j = \sqrt{\sum_{i=1}^n (X_{ji} - W_{ji})^2} \quad (2.4)$$

5. Memilih unit pemenang (indeks j), yaitu unit yang memiliki D_j minimum.
6. Memperbaiki W_{ji} sesuai dengan indeks j dari langkah 5, dengan ketentuan:
 - Apabila $C_j =$ sesuai target kelas, maka menggunakan Persamaan (2.5):

$$W_{ji}(\text{baru}) = W_{ji}(\text{lama}) + \alpha(X_{ji} - W_{ji}(\text{lama})) \quad (2.5)$$
 - Apabila $C_j \neq$ sesuai target kelas, maka menggunakan Persamaan (2.6):

$$W_{ji}(\text{baru}) = W_{ji}(\text{lama}) - \alpha(X_{ji} - W_{ji}(\text{lama})) \quad (2.6)$$
7. Kurangi nilai α di setiap akhir iterasi dengan Persamaan (2.7):

$$\alpha(\text{baru}) = Dec * \alpha(\text{lama}) \quad (2.7)$$

8. Lakukan perulangan hingga memenuhi batas iterasi yang ditentukan.
9. Jika bobot akhir sudah di dapat dari iterasi terakhir maka lakukan testing pada data yang akan di klasifikasikan dengan menggunakan Persamaan (2.4).

2.7 Kombinasi SOM dan LVQ

Kombinasi *SOM* dan *LVQ* diawali dengan pelatihan jaringan *SOM* yang nantinya pada pelatihan *SOM* menghasilkan bobot terlatih. Bobot hasil pelatihan *SOM* tersebut digunakan sebagai bobot awal pada pelatihan *LVQ*. Proses pelatihan pada jaringan *SOM* dilakukan sebagai berikut (Chen, et al., 2012):

1. Lakukan pelatihan pada *SOM* hingga batas yang ditentukan menggunakan Persamaan yang sudah dijelaskan pada subbab 2.5 seperti:
 - a) Melakukan inisialisasi berupa:
 - Inisialisasi vektor bobot (W_{ji}) sebanyak jumlah kelas yang ditentukan.
 - Set nilai pengurang laju pembelajaran (Dec).
 - Set parameter laju pembelajaran (α).
 - Set maksimum iterasi (syarat berhenti).
 - b) Jika syarat berhenti tidak dipenuhi (Salah), Kerjakan langkah c – h.
 - c) Untuk setiap *input* vektor X , kerjakan langkah d – f.
 - d) Untuk setiap indeks j , hitung nilai D_j menggunakan Persamaan (2.1).
 - e) Cari unit pemenang (indeks j), yaitu unit yang memiliki D_j minimum.
 - f) Perbaiki nilai W_{ji} sesuai dengan indeks j dari langkah e menggunakan Persamaan (2.2).
 - g) Ubah (*update*) nilai laju pembelajaran dengan Persamaan (2.3)
 - h) Periksa syarat berhenti.
2. Masukkan semua data latih secara bergantian sesuai proses pada poin b.
3. Lakukan proses kedua hingga syarat berhenti yang ditentukan tercapai (iterasi maksimum).

Lalu dilanjutkan dengan pelatihan jaringan *LVQ* menggunakan vektor bobot hasil pelatihan jaringan *SOM*. Pelatihan dan Persamaan *LVQ* sudah dijelaskan pada subbab 2.6 seperti:

1. Menentukan *input*
2. Menentukan batas iterasi (*epoch*) dan α (*learning rate*)
3. Melakukan perhitungan apabila iterasi $\leq Max$ iterasi dan $\alpha \geq Min\alpha$
4. Menghitung jarak setiap X dengan setiap W menggunakan rumus *Euclidean*, yaitu menggunakan Persamaan (2.4).
5. Memilih jarak terdekat (D_j minimum)
6. Memperbaiki nilai bobot (W) yang memiliki jarak terdekat, dengan ketentuan:
 - Apabila C_j = sesuai target kelas, maka menggunakan Persamaan (2.5).
 - Apabila $C_j \neq$ sesuai target kelas, maka menggunakan Persamaan (2.6).
7. Kurangi nilai α di setiap akhir iterasi dengan Persamaan (2.7).
8. Lakukan perulangan hingga memenuhi batas iterasi yang ditentukan.

9. Jika bobot akhir sudah di dapat dari iterasi terakhir maka lakukan testing pada data yang akan di klasifikasikan dengan menggunakan Persamaan (2.4).

2.8 Nilai Evaluasi

Nilai evaluasi sangat bermanfaat guna membandingkan apakah algoritme yang dipakai atau dikombinasikan pada penelitian ini menghasilkan tingkat akurasi yang lebih baik atau bahkan lebih buruk dibandingkan menggunakan satu metode murni. Pada penelitian ini nilai evaluasi digunakan untuk membandingkan apakah algoritme *Learning Vector Quantization (LVQ)* dengan penentuan bobot awal menggunakan algoritme *Self Organizing Maps (SOM)* menghasilkan tingkat akurasi pengklasifikasian yang lebih baik daripada hanya menggunakan metode *Learning Vector Quantization (LVQ)* saja. Proses evaluasi dilakukan dengan cara melatih data dan membandingkan kesesuaian hasil klasifikasi. Keluaran dari proses evaluasi adalah persentase nilai akurasi.

Penulisan istilah yang ada pada proses perhitungan nilai akurasi adalah sebagai berikut:

NA: Nilai akurasi

$N(t)$: Jumlah berapa banyak data yang benar diklasifikasikan

N : Jumlah seluruh data

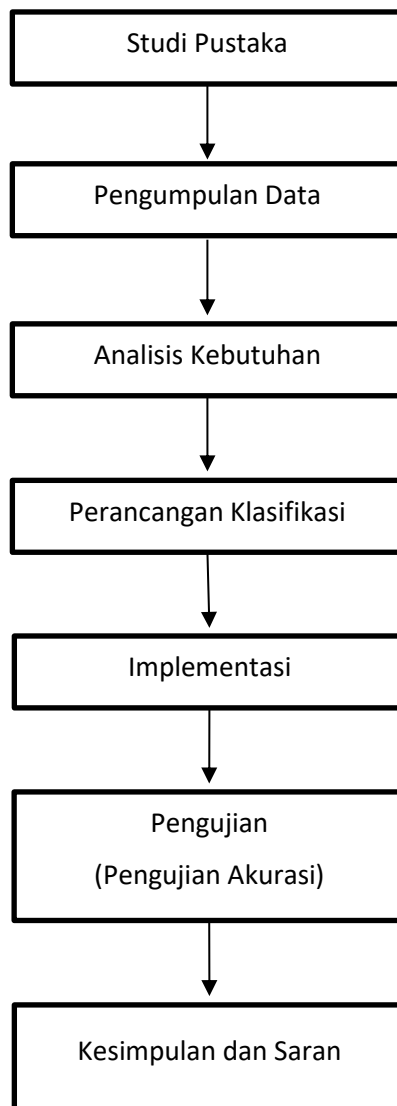
Proses perhitungan nilai akurasi menggunakan Persamaan (2.7) dengan hasil keluaran berupa persentase akurasi sebagai berikut:

$$NA = \frac{N(t)}{N} \times 100\% \quad (2.7)$$

Setelah proses perhitungan dan didapatkan nilai akurasi dari proses pengujian, lalu dilakukan evaluasi dan dipilih nilai dari variabel terbaik hasil dari proses perhitungan akurasi tersebut. Nilai dari variabel yang menghasilkan nilai akurasi terbaik kepada data latih tersebut nantinya yang digunakan untuk mengklasifikasikan data uji masukkan dari *user* dengan harapan memberikan nilai klasifikasi yang terbaik (Putri & Hartati, 2016). Proses evaluasi juga membandingkan hasil perhitungan nilai akurasi terbaik yang didapat jika menggunakan kombinasi algoritme *Learning Vector Quantization (LVQ)* dan *Self Organizing Maps (SOM)* dengan hanya menggunakan algoritme *Learning Vector Quantization (LVQ)*. Pemilihan vektor bobot terbaik dengan cara perhitungan nilai akurasi juga dilakukan pada algoritme *Self Organizing Maps (SOM)* untuk memberikan vektor bobot terbaik pada proses klasifikasi menggunakan algoritme *LVQ*.

BAB 3 METODOLOGI

Metodologi penelitian yang dilakukan dalam penelitian ini melalui beberapa tahapan yaitu studi pustaka, pengumpulan data, analisa kebutuhan, perancangan klasifikasi, implementasi, pengujian, kesimpulan dan saran. Tahapan tersebut diilustrasikan dengan diagram blok metodologi penelitian pada Gambar 3.1.



Gambar 3.1 Diagram Blok Metodologi Penelitian

3.1 Studi Pustaka

Studi Pustaka adalah metode untuk mendapatkan teori pendukung penelitian dan menjadi dasar dalam penelitian ini. Literatur tersebut dapat diperoleh dari buku, jurnal, ebook dan dokumentasi project. Bagian studi literatur ini mencakup teori diantaranya adalah sebagai berikut:

- a. Klasifikasi jenis audio dan musik
- b. Psikologi manusia
- c. Jenis audio dan musik
- d. Algoritme *Learning Vector Quantization*
- e. Algoritme *Self Organizing Maps*
- f. Kombinasi *SOM* dan *LVQ*

3.2 Pengumpulan Data

Pengumpulan data dilakukan dengan wawancara secara langsung atau melalui media online. Target wawancara adalah lingkungan sekitar seperti teman, kerabat, saudara dan yang bersangkutan. Wawancara dilakukan secara langsung jika target wawancara berada pada kota yang sama dan juga melalui media online jika target wawancara berada di kota yang berbeda. Bahan pertanyaan untuk wawancara dirancang sedemikian rupa agar dapat menggambarkan dan mewakili dari keseluruhan data yang dibutuhkan. Hasil dari wawancara dicatat dan dijadikan tabel. Hasil wawancara diolah menjadi biner atau angka dan masukkan ke dalam tabel agar mudah dihitung dan diimplementasikan.

Data yang sudah siap digunakan nantinya akan menjadi data latih dari proses pengklasifikasian yang akan dilakukan menggunakan algoritme *Self Organizing Maps (SOM)* sebagai penentu bobot awal dan *Learning Vector Quantization (LVQ)* sebagai algoritme klasifikasi. Data yang terkumpul terlampir pada Lampiran B, yaitu *dataset 1 (D1)* dengan jumlah data 20, *dataset 2 (D2)* dengan jumlah data 100, *dataset 3 (D3)* dengan jumlah data 31 dan *dataset 4 (D4)* dengan jumlah data 160.

3.3 Analisis Kebutuhan

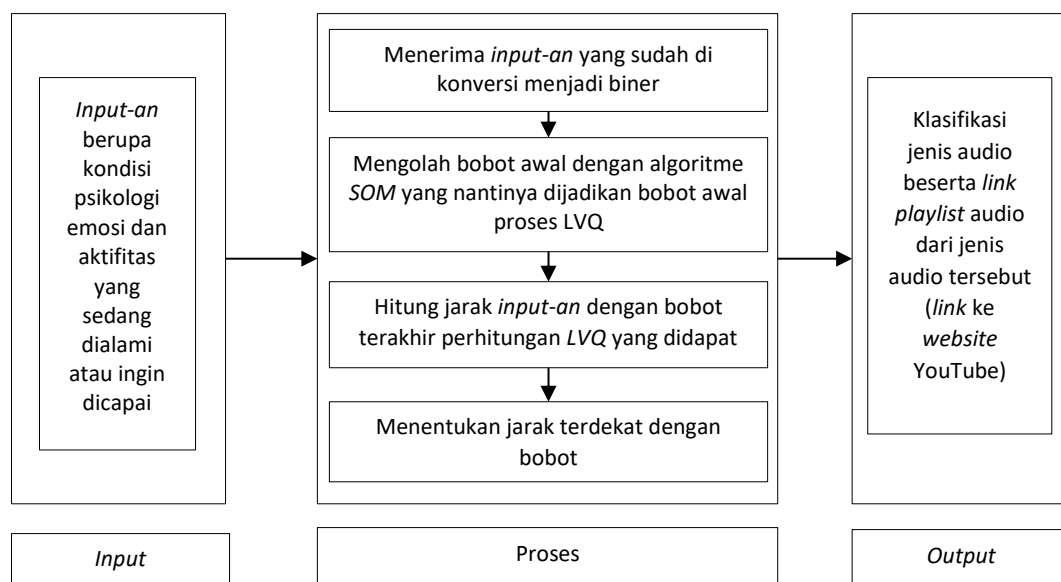
Analisis kebutuhan dilaksanakan untuk mengetahui secara keseluruhan kebutuhan yang harus ada dalam merancang, mengimplementasi dan menguji. Kebutuhan data, meliputi: Data wawancara dan kuesioner yang telah diolah menjadi data yang siap pakai. Kebutuhan minimal yang digunakan dalam perancangan, implementasi dan pengujian adalah sebagai berikut:

1. Kebutuhan Hardware, meliputi:
 - Laptop dengan Prosesor Intel Quad Core / i3
 - RAM 4 GB
2. Kebutuhan Software, meliputi:
 - Windows sebagai sistem operasi
 - Netbean sebagai IDE bahasa pemrograman java
 - Microsoft Office sebagai perhitungan manual dan penempatan data

3.4 Perancangan

Perancangan klasifikasi dilakukan untuk mempermudah memahami dan mengimplementasikan algoritme yang digunakan dalam mengolah masalah. Perancangan dimulai dari mendeskripsikan masalah/objek yang akan diolah, lalu merancang algoritme menggunakan diagram alir guna mempermudah dalam memahami alur kerja dari masing-masing algoritme. Setelah data dan algoritme dirancang, perhitungan manual dilakukan untuk mengetahui apakah data yang sudah diolah dapat diproses menggunakan algoritme yang digunakan. Setelah proses perancangan data, algoritme dan perhitungan manual selesai, perancangan antarmuka dilakukan untuk mempermudah pengguna dalam memasukan data dan menerima hasil/keluaran.

Setelah perancangan telah dibuat, proses klasifikasi dimodelkan sedemikian rupa sesuai urutan proses dari data dimasukan hingga menghasilkan keluaran. Gambaran model perancangan dapat dilihat pada Gambar 3.2.



Gambar 3.2 Model Perancangan Klasifikasi Genre Musik

3.5 Implementasi

Implementasi klasifikasi jenis audio menggunakan algoritme *SOM* dan *LVQ* adalah fase pembentukan klasifikasi yang telah dirancang dan menerapkan hal-hal yang telah diperoleh dalam proses studi literatur. Fase-fase yang ada di dalam implementasi antara lain:

1. Program dibuat dengan bahasa *java* menggunakan *IDE Netbean*
2. Memasukan data penelitian ke dokumen *CSV* sebagai *database* dalam bahasa *java* untuk diolah menjadi informasi yang berguna.
3. Penerapan algoritme *Self Organizing Maps (SOM)* dan *Learning Vector Quantization (LVQ)* dalam program/source code.

3.6 Pengujian

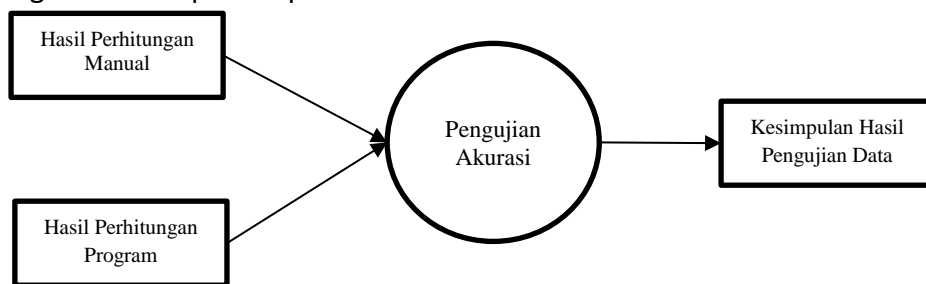
Pengujian klasifikasi yang dilakukan berkaitan dengan pengujian validasi dan akurasi. Tahap validasi berfungsi untuk memastikan pengklasifikasian yang dibuat sesuai dengan perhitungan manual dan tahap akurasi berfungsi untuk mengukur kesesuaian atau akurasi dari klasifikasi yang dibuat. Tahap perhitungan akurasi juga dapat dijadikan tahap pemilihan variabel terbaik. Skenario pengujian ditunjukkan pada Gambar 3.3. Penjelasan pengujian dipaparkan sebagai berikut:

1. Pengujian Validasi

Pengujian validasi dilakukan untuk memastikan pengklasifikasian yang dibuat sesuai dengan perhitungan manual. Pengujian validasi dilakukan dengan membandingkan apakah hasil yang didapat dari perhitungan manual dan perhitungan pada program telah sesuai.

2. Pengujian Akurasi

Pengujian akurasi dilakukan dengan membandingkan hasil klasifikasi dengan dataset, apakah sudah sesuai dengan kelasnya atau belum. Lalu setiap hasil yang sesuai dengan kelasnya akan dihitung dan dibagi dengan keseluruhan data guna mendapatkan persentase akurasi.



Gambar 3.3 Diagram Blok Pengujian Akurasi Sistem

3.7 Kesimpulan dan Saran

Pada penelitian yang diusulkan ini dapat ditarik kesimpulan berdasarkan beberapa uraian dan penjelasan dari bab sebelumnya, bahwa audio terbukti memiliki pengaruh terhadap emosi atau perasaan manusia dan pengaruh tersebut dapat menjadi hal yang menguntungkan seperti menjadi penyemangat suatu aktifitas yang sedang dijalani. Audio dibedakan menjadi berbagai macam jenis atau genre berdasarkan karakteristiknya. Setiap jenis memiliki pengaruh yang berbeda-beda terhadap emosi dan perasaan manusia dalam menjalani aktifitas. Saran disampaikan berdasarkan apa yang telah penulis ketahui terhadap algoritme, pemilihan fitur, nilai akurasi dan juga objek yang dijadikan penelitian. Dari penjabaran diatas dapat dijadikan saran untuk penelitian yang lebih lanjut, yaitu perlu penelitian yang lebih dalam terkait emosi dan perasaan manusia terkait jenis audio dan musik guna menentukan fitur yang lebih baik dan lebih akurat agar fitur yang digunakan dapat mewakili setidaknya masing-masing satu kelas. Ilmu tentang keterkaitan emosi dan perasaan terhadap genre musik juga perlu dikaji lebih dalam untuk memperbaiki hasil klasifikasi.

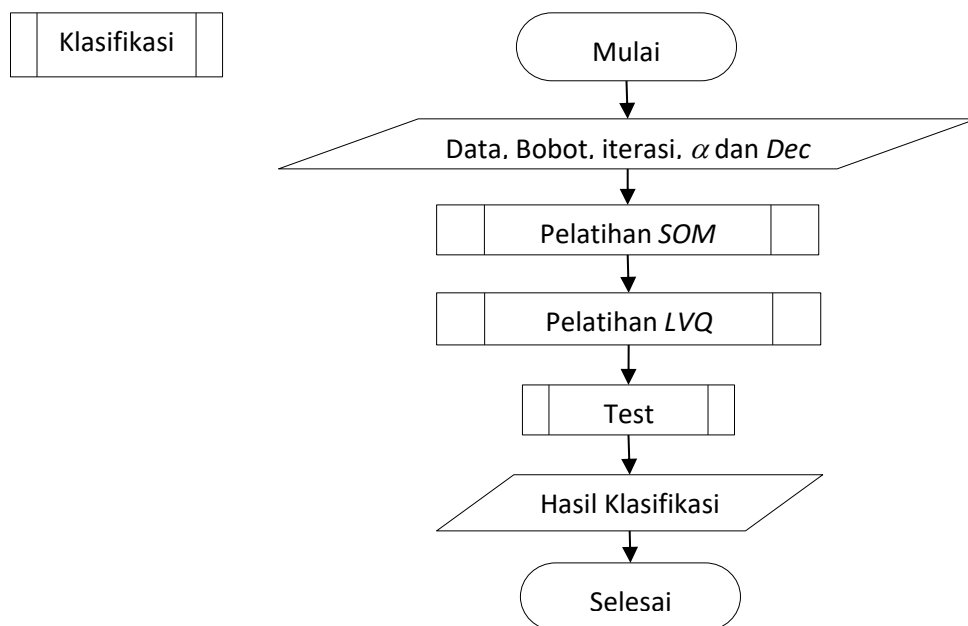
BAB 4 PERANCANGAN

4.1 Deskripsi Masalah

Penelitian yang diusulkan ini akan mencoba menghubungkan antara kondisi psikologi manusia dan jenis kegiatan dengan jenis audio yang cocok didengarkan. Jenis audio diklasifikasikan berdasarkan kondisi psikologi. Harapan dari penelitian ini yaitu dapat menghubungkan kondisi psikologi dengan jenis audio yang cocok didengarkan, mempermudah dan membantu seseorang dalam menentukan jenis audio yang sesuai dengan kondisi. Salah satu jenis kondisi psikologi yang digunakan dalam penelitian ini adalah emosi manusia. Emosi manusia dan jenis kegiatan direpresentasikan menjadi angka biner sebagai fitur. Jenis audio direpresentasikan sebagai kelas atau label. Fitur dan kelas direpresentasikan menjadi satu kesatuan data untuk diolah menggunakan algoritme yang digunakan pada penelitian ini. Deskripsi masalah bertujuan agar memformulasi objek menjadi struktur data yang sesuai dengan algoritme.

4.2 Perancangan Klasifikasi

Pada penelitian ini *Self Organizing Maps (SOM)* bertindak sebagai penentu bobot awal, lalu data dilatih kembali dengan algoritme *Learning Vector Quantization (LVQ)* menggunakan bobot yang sudah dilatih *SOM*. Hasil pelatihan *LVQ* digunakan untuk melakukan proses pengklasifikasian. Proses pengklasifikasian dilakukan dengan cara menghitung jarak data dengan bobot hasil pelatihan *LVQ* menggunakan *Euclidean Distance* yang terdapat pada fungsi Test. Kombinasi algoritme *SOM* dan *LVQ* untuk proses klasifikasi dirancang sesuai urutan prosesnya dalam bentuk diagram alir pada Gambar 4.1.



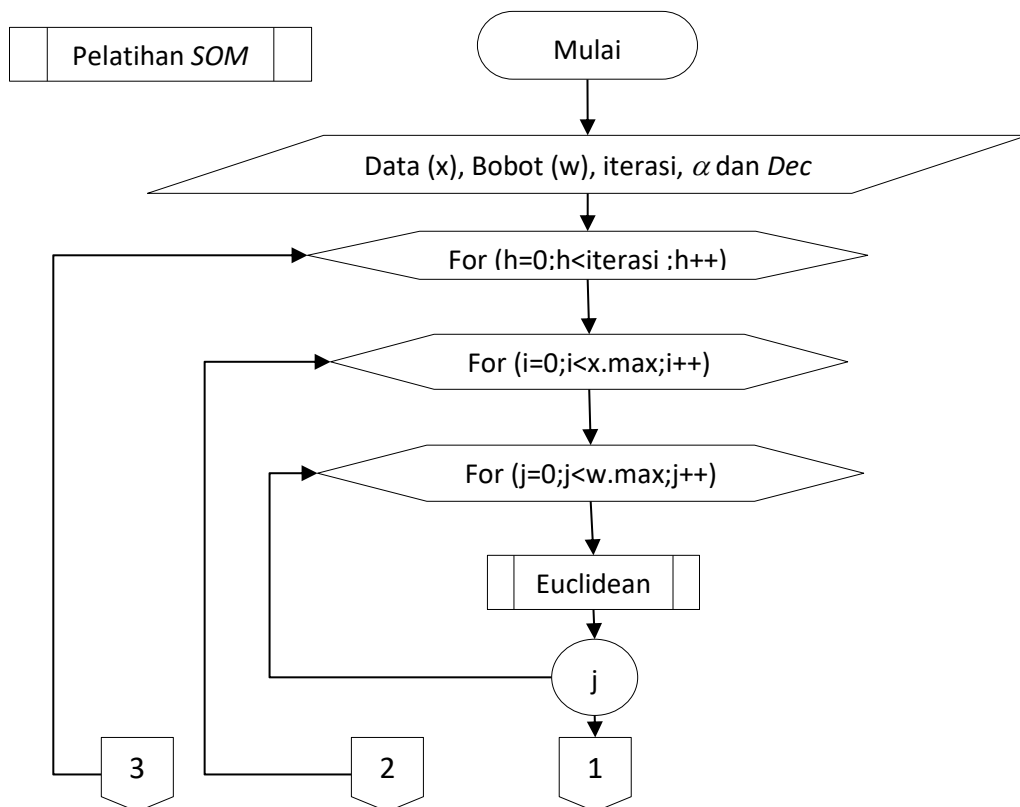
Gambar 4.1 Diagram Alir Klasifikasi

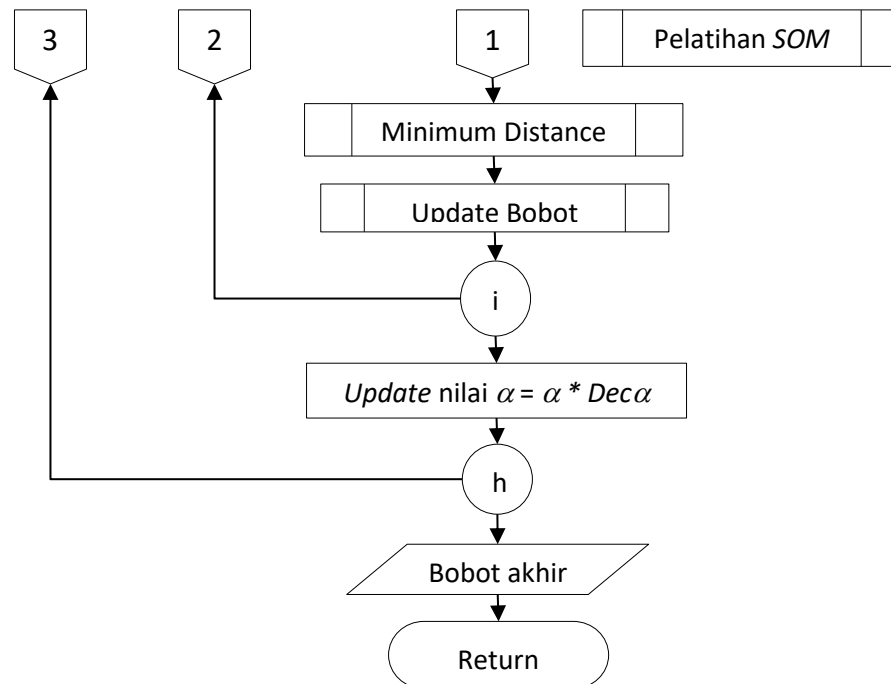
4.3 Perancangan Algoritme

Dalam perancangan algoritme, dipaparkan urutan proses dan contoh manualisasi dari masing-masing algoritme yang dipakai. Pada penelitian ini algoritme yang digunakan adalah *Self Organizing Maps (SOM)* sebagai penentu bobot awal, lalu *Learning Vector Quantization (LVQ)* untuk melakukan proses pengklasifikasian. Perancangan dari tiap algoritme beserta fungsi-fungsi penyusun algoritme dijelaskan dalam bentuk diagram alir pada masing-masing subbab.

4.3.1 Self Organizing Maps

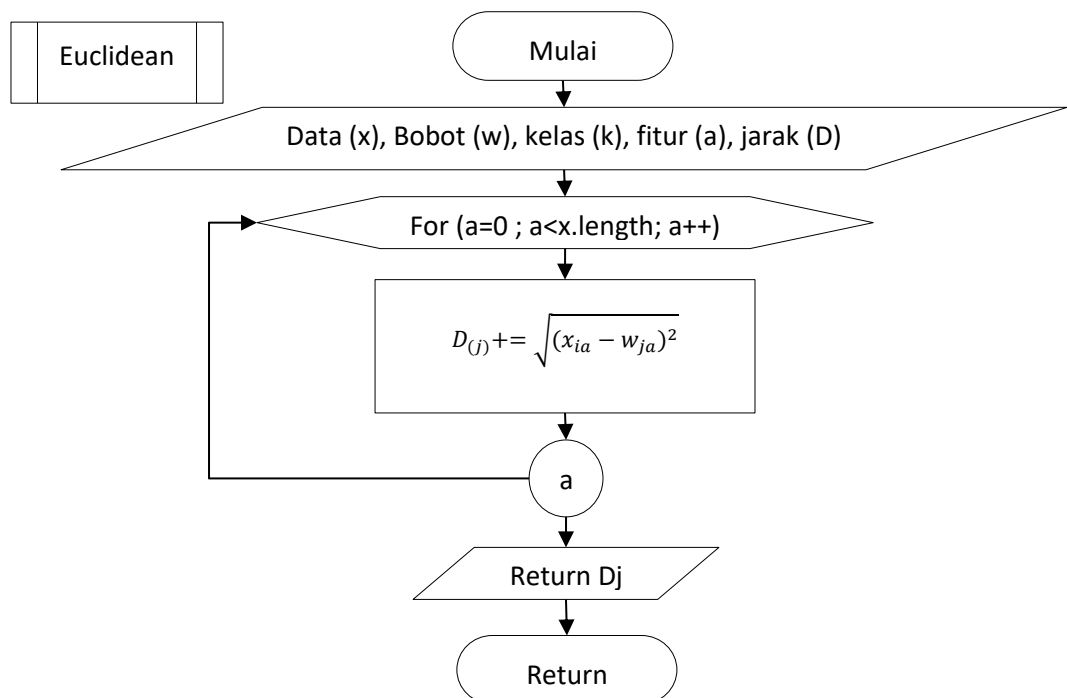
Proses dari pelatihan *SOM* dimulai dari menentukan data latih, Inisialisasi bobot dari data latih, set nilai pengurang laju pembelajaran (*Dec*), set parameter laju pembelajaran (α), set maksimum iterasi. Proses pertama adalah menghitung jarak satu vektor data x dengan setiap vektor bobot w menggunakan rumus *Euclidean* yaitu menggunakan Persamaan (2.1) dipaparkan dalam Gambar 4.3. Proses kedua adalah mencari nilai jarak minimum menggunakan fungsi *Minimum Distance* yang dijelaskan pada Gambar 4.4. Unit pemenang (indeks j) yaitu unit yang memiliki D_j minimum (*Minimum Distance*). Proses ketiga adalah *update* semua nilai W_{ji} berdasarkan D_j menggunakan Persamaan (2.2) dipaparkan dalam Gambar 4.5. Lakukan proses tersebut hingga semua vektor data x sudah dihitung jaraknya dengan vektor bobot w , lalu *update* nilai laju pembelajaran menggunakan Persamaan (2.3) dan periksa syarat berhenti. Lakukan semua proses tersebut hingga iterasi yang ditentukan. Proses dari pelatihan *SOM* dirancang dalam diagram alir pada Gambar 4.2.





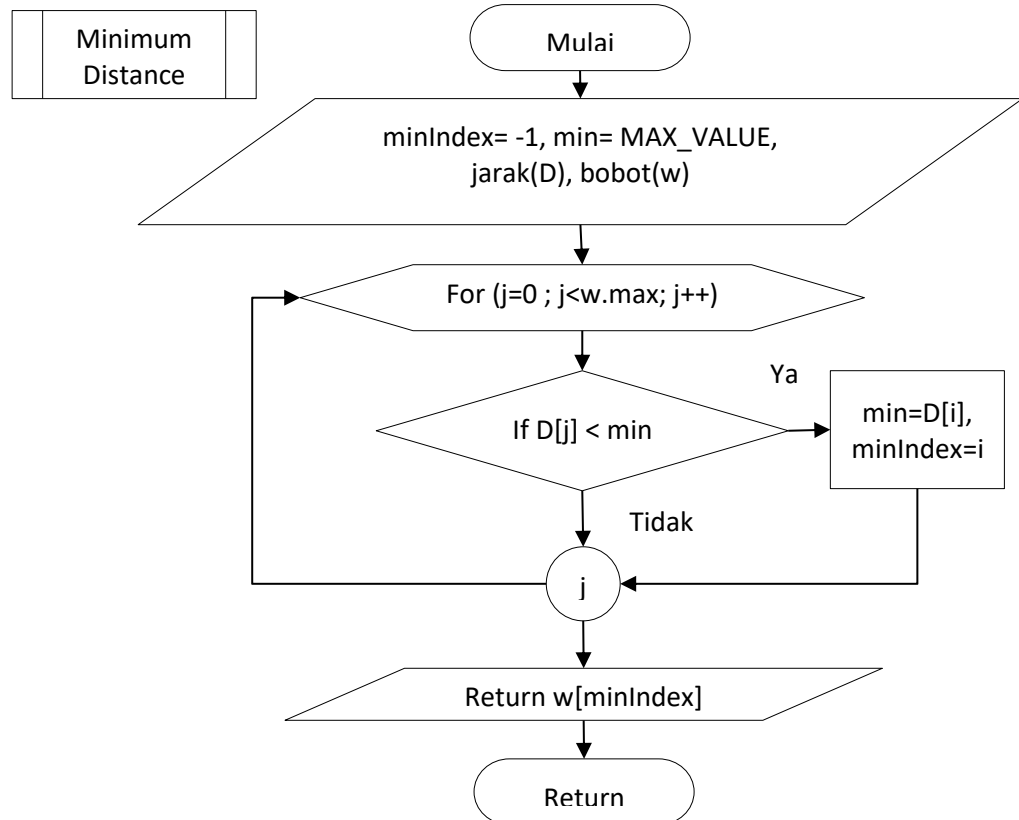
Gambar 4.2 Diagram Alir Pelatihan SOM

Gambar 4.3 adalah diagram alir proses perhitungan jarak *Euclidean*. Proses perhitungan jarak dimulai dengan menentukan vektor data dan vektor bobot yang akan dihitung nilai jaraknya, lalu lakukan proses perhitungan jarak menggunakan Persamaan (2.1). Setiap vektor data dan vektor bobot memiliki jumlah fitur yang sama, yaitu 11 fitur. Nilai hasil perhitungan jarak antara vektor data dan vektor bobot disimpan pada variabel D_j .



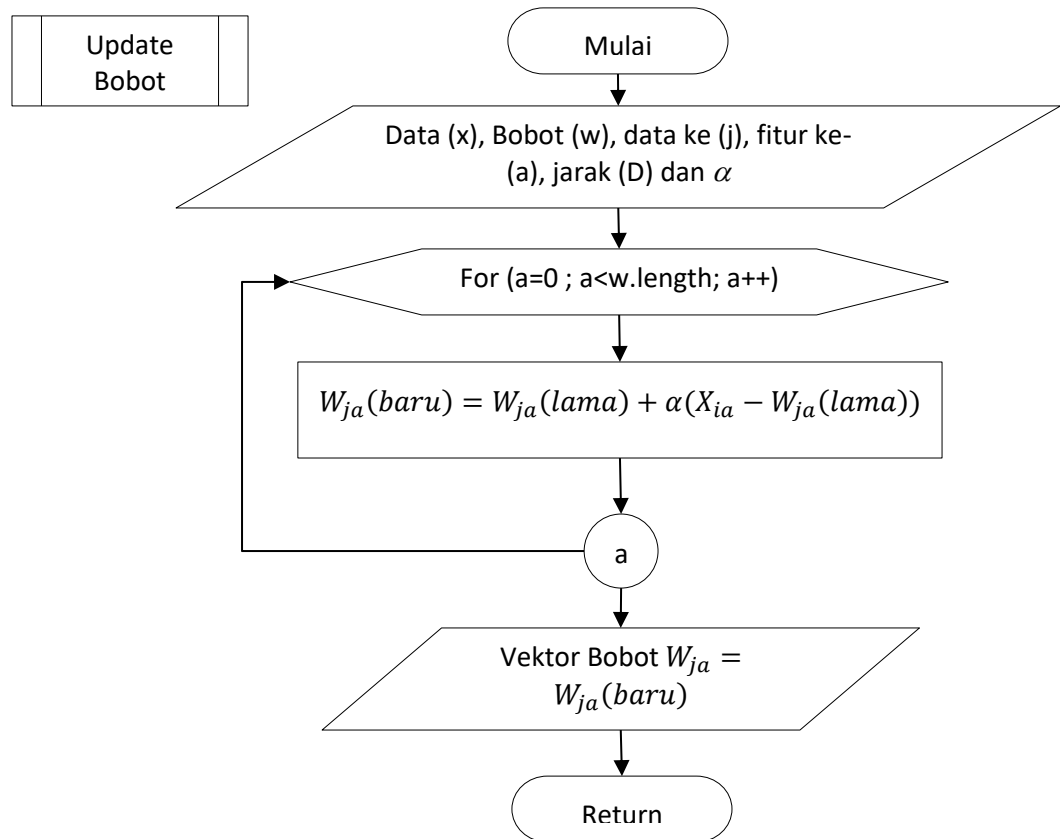
Gambar 4.3 Diagram Alir Perhitungan Jarak

Gambar 4.4 adalah diagram alir proses pemilihan nilai jarak terkecil. Proses pertama yang dilakukan adalah menentukan variabel yang mewakili nilai besar yaitu minIndex dan nilai kecil yaitu min. Kedua variabel tersebut berfungsi menjadi pembanding untuk mencari jarak yang memiliki nilai terkecil. Proses selanjutnya adalah perbandingan menggunakan logika IF, yaitu jika nilai jarak pada indeks ke- i lebih kecil dari nilai min maka simpan nilai vektor D indeks ke- j menjadi nilai terkecil dan juga indeks terkecil, jika tidak maka abaikan. Lakukan proses tersebut sebanyak vektor jarak. Jika semua proses sudah selesai, simpan indeks vektor terkecil untuk digunakan sebagai indeks dari vektor bobot.



Gambar 4.4 Diagram Alir Jarak Terdekat

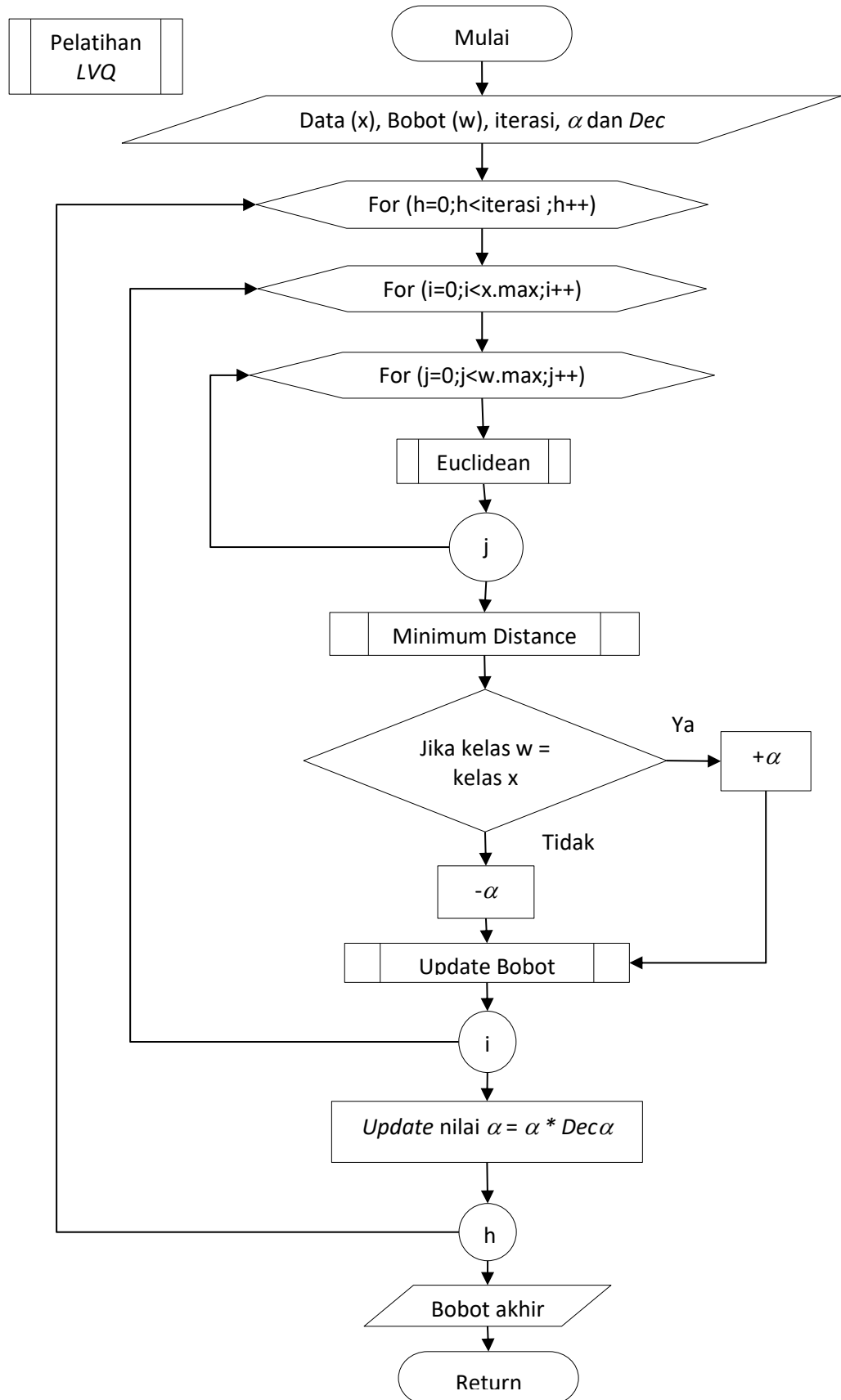
Gambar 4.5 adalah diagram alir proses *update* bobot. Proses pertama adalah menentukan bobot yang akan diperbaiki/*update*, data yang berhubungan dengan bobot, indeks fitur dan laju pembelajaran (α). Proses selanjutnya adalah melakukan *update* nilai bobot sebanyak jumlah indeks fiturnya (11) menggunakan Persamaan (2.2). Setelah selesai, simpan semua nilai bobot yang telah diperbaiki kedalam variabel bobot yang bersangkutan.



Gambar 4.5 Diagram Alir *Update Bobot*

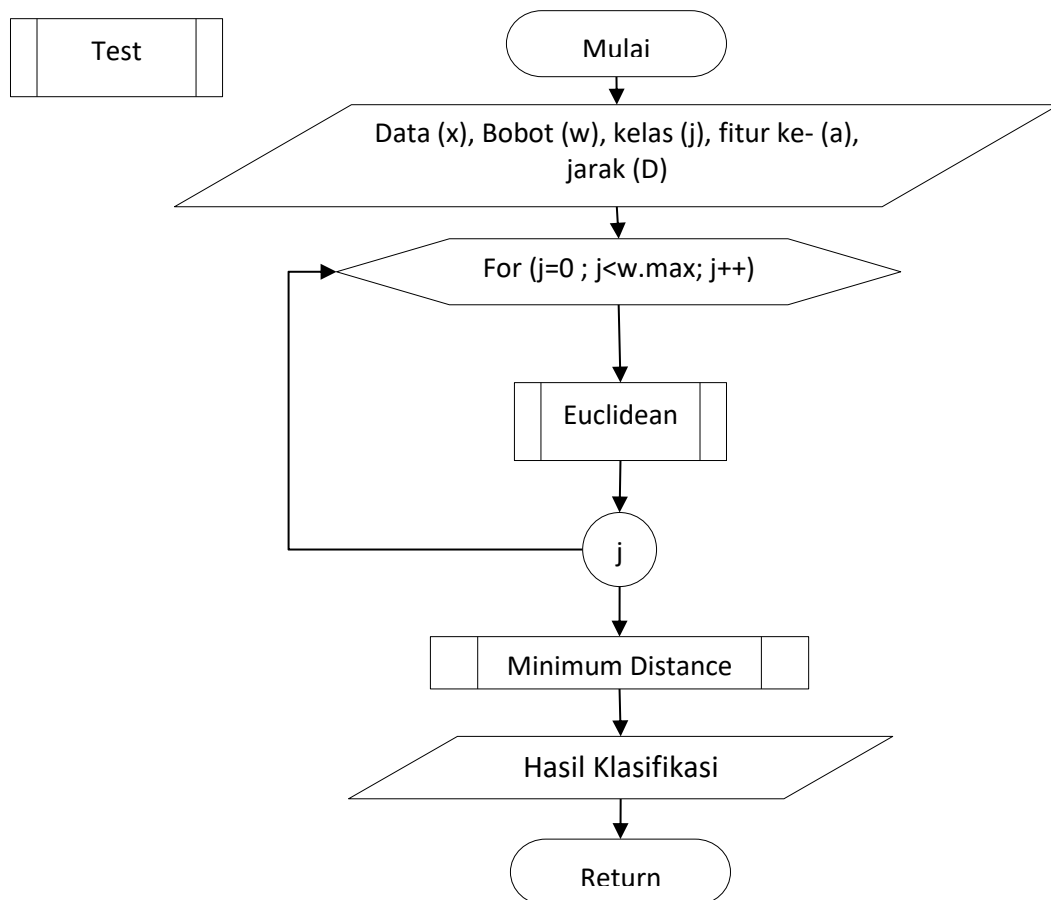
4.3.2 Learning Vector Quantization

Proses dari pelatihan *Learning Vector Quantization* (LVQ) dimulai dari menentukan data latih, menentukan bobot yang digunakan, set nilai pengurang laju pembelajaran (*Dec*), set parameter laju pembelajaran (α), set maksimum iterasi. Proses pertama adalah menghitung jarak satu vektor data x dengan setiap vektor bobot w menggunakan rumus *Euclidean* yaitu menggunakan Persamaan (2.4) dipaparkan dalam Gambar 4.3. Proses kedua adalah mencari nilai jarak minimum menggunakan fungsi *Minimum Distance* yang dijelaskan pada Gambar 4.4. Unit pemenang (indeks j) yaitu unit yang memiliki D_j minimum (*Minimum Distance*). Proses ketiga adalah *update* semua nilai W_{ji} berdasarkan D_j menggunakan Persamaan (2.5) jika kelas dari vektor data sesuai dengan vektor bobot terpilih, atau menggunakan Persamaan (2.6) jika tidak sesuai. Proses *update* nilai bobot dipaparkan dalam Gambar 4.5. Lakukan proses tersebut hingga semua vektor data x sudah dihitung jaraknya dengan vektor bobot w , lalu *update* nilai laju pembelajaran menggunakan Persamaan (2.7) dan periksa syarat berhenti. Lakukan semua proses tersebut hingga iterasi yang ditentukan. Proses dari pelatihan LVQ dirancang dalam diagram alir pada Gambar 4.6.



Gambar 4.6 Diagram Alir Pelatihan LVQ

Gambar 4.7 adalah diagram alir yang menjelaskan proses *testing*. *Testing* adalah proses penentuan/pengecekan vektor data yang di proses memiliki nilai jarak terdekat dengan salah satu vektor bobot. Proses pertama adalah menentukan vektor data, vektor bobot, variabel jarak dan variabel kelas. Proses selanjutnya adalah menghitung nilai jarak antara vektor data dengan vektor bobot menggunakan *Euclidean Distance* seperti pada Gambar 4.3. Jika sudah dihitung jarak antara vektor data kepada seluruh vektor bobot maka tentukan jarak terkecil menggunakan *Minimum Distance* seperti pada Gambar 4.4. Simpan indeks jarak terkecil.



Gambar 4.7 Diagram Alir Test Data

4.3.3 Contoh Manualisasi Pelatihan *SOM-LVQ*

Setelah dirancang alur proses dari pelatihan *SOM*, perancangan *SOM* diperjelas dalam bentuk manualisasi. Terdapat total 20 data latih, 11 fitur yang ada pada data latih yaitu dari konsentrasi hingga istirahat (X1 hingga X11) dan 4 kelas. Tabel 4.1 menampilkan potongan data latih dari Lampiran B.1.

Tabel 4.1 Contoh Data Latih

DATA	FITUR					KELAS
	X1	X2	...	X10	X11	
1	1	0		0	0	klasik
2	1	1		0	0	klasik
3	1	1		0	0	klasik
4	1	1		0	1	klasik
5	0	1		0	0	klasik
6	0	1		0	1	Murottal
7	0	1		0	1	Murottal
8	0	1		0	1	Murottal
9	0	1		0	1	Murottal
10	0	1		0	1	Murottal
11	0	0		1	0	rock
12	0	0		1	0	rock
13	0	0		1	0	rock
14	0	0		0	1	rock
15	0	0		1	0	rock
16	0	0		0	1	pop
17	0	0		0	1	pop
18	0	0		0	1	pop
19	0	1		0	1	pop
20	0	1		0	0	pop

Setelah data telah tersusun, kerjakan langkah-langkah pelatihan *SOM* seperti pada subbab 2.5. Urutan proses pelatihan antara lain:

➤ Iterasi 1

1. Inisialisasi bobot awal, α , Dec dan iterasi:

- Data 1, 6, 11, 16 dipilih secara acak (sebanyak jumlah kelas) dan menjadi bobot awal ($X_{1i} = W_{1i}$, $X_{6i} = W_{6i}$ dan seterusnya hingga data 16).
- α di tentukan dalam rentang 0 hingga 1, pada contoh kasus ini $\alpha=0,1$
- Dec adalah pengurang α , pada contoh kasus ini $Dec=0,1$
- Iterasi menjadi kondisi berhenti, pada contoh kasus ini iterasi=2

2. Menghitung jarak setiap X dengan setiap W menggunakan rumus *Euclidean*, yaitu menggunakan Persamaan (2.1). Contoh data yang akan dihitung jaraknya adalah data 2:

- Jarak antara data 2 (X_{2i}) dengan bobot 1 (W_{1i}):

$$D_1 = \sqrt{\sum_{i=1}^{11} ((X_{2,i} - W_{1,i}))^2}$$

$$D_1 = \sqrt{\sum_{i=1}^{11} ((1 - 1) + (1 - 0) + (0 - 0) + (0 - 0) + (0 - 0) + \dots + (0 - 0))^2}$$

$$D_1 = 1$$

- Jarak antara data 2 (X_{2i}) dengan bobot 2 (W_{2i}):

$$D_2 = \sqrt{\sum_1^{11}((X_{2,1} - W_{2,1}) + (X_{2,2} - W_{2,2}) + \dots + (X_{2,11} - W_{2,11}))^2}$$

$$D_2 = \sqrt{\sum_1^{11}((1 - 0) + (1 - 1) + (0 - 0) + (0 - 0) + (0 - 0) + \dots + (0 - 1))^2}$$

$$D_2 = 1,7321$$

- Jarak antara data 2 (X_{2i}) dengan bobot 3 (W_{3i}):

$$D_3 = \sqrt{\sum_1^{11}((X_{2,1} - W_{3,1}) + (X_{2,2} - W_{3,2}) + \dots + (X_{2,11} - W_{3,11}))^2}$$

$$D_3 = \sqrt{\sum_1^{11}((1 - 0) + (1 - 0) + (0 - 0) + (0 - 0) + (0 - 1) + \dots + (0 - 0))^2}$$

$$D_3 = 2,2361$$

- Jarak antara data 2 (X_{2i}) dengan bobot 4 (W_{4i}):

$$D_4 = \sqrt{\sum_1^{11}((X_{2,1} - W_{4,1}) + (X_{2,2} - W_{4,2}) + \dots + (X_{2,11} - W_{4,11}))^2}$$

$$D_4 = \sqrt{\sum_1^{11}((1 - 0) + (1 - 0) + (0 - 0) + (0 - 0) + (0 - 0) + \dots + (0 - 1))^2}$$

$$D_4 = 2,2361$$

3. Tentukan *neuron* pemenang (D_j minimum):

- D_1 merupakan *Distance* terdekat.

4. *Update* bobot pada D_j minimum:

- D_j minimum = $D_1 = W_{1i}$
- *Update* bobot W_{1i} menggunakan Persamaan (2.2):
 $W_{1,1}(\text{baru}) = 1 + 0,1 (1 - 1) = 1$
 $W_{1,2}(\text{baru}) = 0 + 0,1 (1 - 0) = 0,1$

...

$$W_{1,11}(\text{baru}) = 0 + 0,1 (0 - 0) = 0$$

5. Ulangi proses 2 sampai 4 kepada seluruh data.
6. *Update* laju pembelajaran α setiap telah melalui 1 iterasi menggunakan Persamaan (2.3):
 $\alpha(\text{baru}) = 0,1 * 0,1$
 $\alpha(\text{baru}) = 0,01$
7. Periksa syarat berhenti (iterasi).
8. Hasil dari perhitungan jarak pada iterasi 1 dipaparkan pada Tabel 4.2 dan hasil perbaikan bobot pada akhir iterasi 1 ditampilkan pada Tabel 4.3.

Tabel 4.2 Hasil Perhitungan Jarak Iterasi 1 *SOM*

Data	Jarak Iterasi 1 <i>SOM</i>			
	d1	d2	d3	d4
2	1	1.732051	2.236068	2.236068
3	1.345362	2	2.44949	2
4	1.63282	1	2.236068	1.732051
5	1.290775	1.417745	2	2
7	2.085555	1.004988	2.236068	1.732051
8	2.041945	1.009009	2.236068	1
9	2.273662	1.348369	2.44949	1.345362

Tabel 4.2 Hasil Perhitungan Jarak Iterasi 1 SOM (Lanjutan)

Data	Jarak Iterasi 1 SOM			
	d1	d2	d3	d4
10	2.085555	1.009009	1.732051	1.63282
12	2.211683	2.196511	1	2.246353
13	2.211683	2.196511	1.004988	2.246353
14	1.972699	1.3508	1.420598	1.43042
15	2.42725	2.340508	1.279883	2.458882
17	2.170608	1.641943	2.251413	0.919837
18	2.211683	1.641943	2.251413	1.638701
19	2.041945	1.028579	2.251413	0.899626
20	1.532821	1.701757	2.251413	1.996936

Tabel 4.3 Hasil Perbaikan Bobot Iterasi 1 SOM

	BOBOT AKHIR ITERASI 1 SOM										
w1	0.81	0.3439	0	0	0	0	0.081	0.1	1	0	0
w2	0.0729	0.9	0	0	0.19	0	0	0.081	0	0	1
w3	0	0	0.19	0	1	0.181	0	0	0	1	0
w4	0	0.23851	0	0.09	0	0	0.91	0.2439	0	0	1

➤ Iterasi 2

1. Menghitung jarak setiap X dengan setiap W menggunakan rumus *Euclidean*, yaitu menggunakan Persamaan (2.1). Contoh data yang akan dihitung jaraknya adalah data 2, tetapi dibandingkan dengan bobot yang sudah didapat dari iterasi 1:

- Jarak antara data 2 (X_{2i}) dengan bobot 1 (W_{1i}):

$$D_1 = \sqrt{\sum_{i=1}^{11} ((X_{2,1} - W_{1,1}) + (X_{2,2} - W_{1,2}) + \dots + (X_{2,11} - W_{1,11}))^2}$$

$$D_1 = \sqrt{\sum_{i=1}^{11} ((1 - 0,81) + (1 - 0,34) + (0 - 0) + (0 - 0) + (0 - 0) + \dots + (0 - 0))^2}$$

$$D_1 = 0,695$$

- Jarak antara data 2 (X_{2i}) dengan bobot 2 (W_{2i}):

$$D_2 = \sqrt{\sum_{i=1}^{11} ((X_{2,1} - W_{2,1}) + (X_{2,2} - W_{2,2}) + \dots + (X_{2,11} - W_{2,11}))^2}$$

$$D_2 = \sqrt{\sum_{i=1}^{11} ((1 + 0,07) + (1 - 0,9) + (0 - 0) + (0 - 0) + (0 - 0,19) + \dots + (0 - 1))^2}$$

$$D_2 = 1,7065$$

- Jarak antara data 2 (X_{2i}) dengan bobot 3 (W_{3i}):

$$D_3 = \sqrt{\sum_{i=1}^{11} ((X_{2,1} - W_{3,1}) + (X_{2,2} - W_{3,2}) + \dots + (X_{2,11} - W_{3,11}))^2}$$

$$D_3 = \sqrt{\sum_1^{11}((1-0) + (1-0) + (0-0,19) + (0-0) + (0-1) + \dots + (0-0))^2}$$

$$D_3 = 2,2514$$

- Jarak antara data 2 (X_{2i}) dengan bobot 4 (W_{4i}):

$$D_4 = \sqrt{\sum_1^{11}((X_{2,1} - W_{4,1}) + (X_{2,2} - W_{4,2}) + \dots + (X_{2,11} - W_{4,11}))^2}$$

$$D_4 = \sqrt{\sum_1^{11}((1+0) + (1-0,24) + (0-0,09) + (0-0) + (0-0) + \dots + (0-1))^2}$$

$$D_4 = 2,1156$$

2. Tentukan *neuron* pemenang (D_j minimum):

- D_1 merupakan *Distance* terdekat.

3. *Update* bobot pada D_j minimum:

- D_j minimum = $D_1 = W_{1i}$
- *Update* bobot W_{1i} menggunakan Persamaan (2.2):
 $W_{1,1}(\text{baru}) = 0,81 + 0,01 (1 - 0,81) = 0,8119$
 $W_{1,2}(\text{baru}) = 0,34 + 0,01 (1 - 0,34) = 0,3505$

...

$$W_{1,11}(\text{baru}) = 0 + 0,01 (0 - 0) = 0$$

4. Ulangi proses 2 sampai 4 kepada seluruh data.
5. *Update* laju pembelajaran α setiap telah melalui 1 iterasi menggunakan Persamaan (2.3):
 $\alpha(\text{baru}) = 0,1 * 0,01$
 $\alpha(\text{baru}) = 0,001$
6. Periksa syarat berhenti (iterasi). Hasil dari perhitungan jarak pada iterasi 2 ditampilkan pada Tabel 4.4 dan hasil perbaikan bobot pada iterasi 2 ditampilkan pada Tabel 4.5.

Tabel 4.4 Hasil Perhitungan Jarak Iterasi 2 *SOM*

Data	Jarak Iterasi 2 <i>SOM</i>			
	d1	d2	d3	d4
2	0.695074	1.70651	2.251413	2.115551
3	1.145921	1.977922	2.463506	1.91195
4	1.57028	0.955079	2.251413	1.573389
5	1.045629	1.4347	2.017142	1.864284
7	1.969143	0.947621	2.251413	1.409877
8	1.973473	1.029024	2.251413	0.809663
9	2.168072	1.371173	2.463506	1.076841
10	2.017816	0.828525	1.751816	1.56786
12	1.672859	2.116302	0.84075	2.228343
13	2.190515	2.116302	0.833651	2.228343
14	1.948937	1.216032	1.439994	1.401968
15	2.407978	2.334375	1.143043	2.44244

Tabel 4.4 Hasil Perhitungan Jarak Iterasi 2 SOM (Lanjutan)

Data	Jarak Iterasi 2 SOM			
	d1	d2	d3	d4
17	2.104117	1.636739	2.253866	0.802395
18	2.145763	1.636739	2.253866	1.510536
19	1.973473	1.032595	2.253866	0.808007
20	1.370228	1.700304	2.253866	1.981465

Tabel 4.5 Hasil Perbaikan Bobot Iterasi 2 SOM

	BOBOT AKHIR ITERASI 2 SOM										
w1	0.797	0.369	0	0	0	0	0.087	0.106	1	0	0
w2	0.079	0.89	0	0	0.2024	0	0	0.0876	0	0	1
w3	0	0	0.204	0	1	0.195	0	0	0	1	0
w4	0	0.2561	0	0.0954	0	0	0.9045	0.26135	0	0	1

Setelah bobot akhir dari pelatihan SOM telah didapatkan pada Tabel 4.5, kerjakan langkah-langkah pelatihan LVQ seperti pada subbab 2.6 dengan bobot yang telah didapatkan dari pelatihan SOM iterasi kedua. Urutan proses pelatihan antara lain:

➤ Iterasi 1

1. Inisialisasi α , Dec dan iterasi:

- α di tentukan dalam rentang 0 hingga 1, pada contoh kasus ini $\alpha=0,1$
- Dec adalah pengurang α , pada contoh kasus ini Dec=0,1
- Iterasi menjadi kondisi berhenti, pada contoh kasus ini iterasi=2

2. Menghitung jarak setiap X dengan setiap W menggunakan rumus Euclidean, yaitu menggunakan Persamaan (2.4). Contoh data yang akan dihitung jaraknya adalah data 2:

- Jarak antara data 2 (X_{2i}) dengan bobot 1 (W_{1i}):

$$D_1 = \sqrt{\sum_{i=1}^{11} ((X_{2,1} - W_{1,1}) + (X_{2,2} - W_{1,2}) + \dots + (X_{2,11} - W_{1,11}))^2}$$

$$D_1 = \sqrt{\sum_{i=1}^{11} ((1 - 0,79) + (1 - 0,37) + (0 - 0) + (0 - 0) + (0 - 0) + \dots + (0 - 0))^2}$$

$$D_1 = 0,676$$

- Jarak antara data 2 (X_{2i}) dengan bobot 2 (W_{2i}):

$$D_2 = \sqrt{\sum_{i=1}^{11} ((X_{2,1} - W_{2,1}) + (X_{2,2} - W_{2,2}) + \dots + (X_{2,11} - W_{2,11}))^2}$$

$$D_2 = \sqrt{\sum_{i=1}^{11} ((1 - 0,79) + (1 - 0,89) + (0 - 0) + (0 - 0) + (0 - 0,2) + \dots + (0 - 1))^2}$$

$$D_2 = 1,7049$$

- Jarak antara data 2 (X_{2i}) dengan bobot 3 (W_{3i}):

$$D_3 = \sqrt{\sum_{i=1}^{11} ((X_{2,1} - W_{3,1}) + (X_{2,2} - W_{3,2}) + \dots + (X_{2,11} - W_{3,11}))^2}$$

$$D_3 = \sqrt{\sum_1^{11} ((1-0) + (1-0) + (0-0,2) + (0-0) + (0-1) + \dots + (0-0))^2}$$

$$D_3 = 2,2539$$

- Jarak antara data 2 (X_{2i}) dengan bobot 4 (W_{4i}):

$$D_4 = \sqrt{\sum_1^{11} ((X_{2,1} - W_{4,1}) + (X_{2,2} - W_{4,2}) + \dots + (X_{2,11} - W_{4,11}))^2}$$

$$D_4 = \sqrt{\sum_1^{11} ((1-0) + (1-0,26) + (0-0) + (0-0,09) + (0-0) + \dots + (0-1))^2}$$

$$D_4 = 2,1093$$

3. Tentukan *neuron* pemenang (D_j minimum):
 - D_1 merupakan *Distance* terdekat.
4. *Update* bobot pada D_j minimum:
 - D_j minimum = $D_1 = W_{1i}$
 - Cek apakah kelas yang dimiliki data sesuai dengan kelas yang dimiliki bobot. Data 2 dan bobot 1 memiliki kelas yang sama, yaitu Klasik maka update bobot menggunakan Persamaan (2.5). Jika kelas yang dimiliki data dan bobot tidak sama maka gunakan Persamaan (2.6).
 - *Update* bobot W_{1i} menggunakan persamaan (2.5):

$$W_{1,1}(\text{baru}) = 0,7976 + 0,1 (1 - 0,7976) = 0,8178$$

$$W_{1,2}(\text{baru}) = 0,3698 + 0,1 (1 - 0,3698) = 0,4328$$

$$\dots$$

$$W_{1,11}(\text{baru}) = 0 + 0,1 (0 - 0) = 0$$
5. Ulangi proses 2 sampai 4 kepada seluruh data.
6. *Update* laju pembelajaran α setiap telah melalui 1 iterasi menggunakan persamaan (2.7):

$$\alpha(\text{baru}) = 0,1 * 0,1$$

$$\alpha(\text{baru}) = 0,01$$
7. Periksa syarat berhenti (iterasi). Hasil dari perhitungan jarak pada iterasi 1 akan ditampilkan pada Tabel 4.6 dan hasil perbaikan bobot pada iterasi 1 akan ditampilkan pada Tabel 4.7.

Tabel 4.6 Hasil Perhitungan Jarak Iterasi 1 *SOM-LVQ*

Data	Jarak Iterasi 1 <i>SOM-LVQ</i>			
	d1	d2	d3	d4
2	0.676054	1.704948	2.253868	2.109255
3	1.101149	1.976575	2.465749	1.907867
4	1.524487	0.952286	2.253868	1.564915
5	0.998085	1.439662	2.019881	1.857137
7	1.911075	0.938034	2.253868	1.387861
8	1.870609	1.042196	2.253868	0.799974
9	2.084343	1.308711	2.465749	1.095111
10	1.951128	0.827901	1.75497	1.628678

Tabel 4.6 Hasil Perhitungan Jarak Iterasi 1 *SOM-LVQ* (Lanjutan)

Data	Jarak Iterasi 1 <i>SOM-LVQ</i>			
	d1	d2	d3	d4
12	2.210898	2.089526	0.830132	2.202806
13	2.210898	2.089526	0.861491	2.202806
14	1.971819	1.168812	1.460178	1.361013
15	2.426534	2.377605	1.051003	2.419164
17	2.103724	1.643179	2.284621	0.80677
18	2.175632	1.643179	2.284621	1.453001
19	1.870609	1.038137	2.284621	1.026192
20	1.285382	1.645604	2.284621	1.961415

Tabel 4.7 Hasil Perbaikan Bobot Iterasi 1 *SOM-LVQ*

	BOBOT AKHIR ITERASI 1 <i>SOM-LVQ</i>										
w1	0.8276 95	0.4946 43	0	0	0	0	0.1692 46	- 0.0149 2	1	0	0
w2	- 0.0109 9	0.9960 11	0	0	0.2083 72	0	0	0.1848 57	0	0	1
w3	0	0	0.3389 35	0	1	0.3234 47	0	0	0	1	0
w4	0	0.1728 13	0	0.174 24	0	0	0.8257 6	0.3286 78	0	0	1

➤ Iterasi 2

1. Menghitung jarak setiap X dengan setiap W menggunakan rumus *Euclidean*, yaitu menggunakan Persamaan (2.4). Contoh data yang akan dihitung jaraknya adalah data 2:

- Jarak antara data 2 (X_{2i}) dengan bobot 1 (W_{1i}):

$$D_1 = \sqrt{\sum_{i=1}^{11} ((X_{2,1} - W_{1,1}) + (X_{2,2} - W_{1,2}) + \dots + (X_{2,11} - W_{1,11}))^2}$$

$$D_1 = \sqrt{\sum_{i=1}^{11} ((1 - 0,83) + (1 - 0,49) + (0 - 0) + (0 - 0) + (0 - 0) + \dots + (0 - 0))^2}$$

$$D_1 = 0,56$$

- Jarak antara data 2 (X_{2i}) dengan bobot 2 (W_{2i}):

$$D_2 = \sqrt{\sum_{i=1}^{11} ((X_{2,1} - W_{2,1}) + (X_{2,2} - W_{2,2}) + \dots + (X_{2,11} - W_{2,11}))^2}$$

$$D_2 = \sqrt{\sum_1^{11}((1 - 0,01) + (1 - 0,99) + (0 - 0) + (0 - 0) + (0 - 0,21) + \dots + (0 - 1))^2}$$

$$D_2 = 1,7606$$

- Jarak antara data 2 (X_{2i}) dengan bobot 3 (W_{3i}):

$$D_3 = \sqrt{\sum_1^{11}((X_{2,1} - W_{3,1}) + (X_{2,2} - W_{3,2}) + \dots + (X_{2,11} - W_{3,11}))^2}$$

$$D_3 = \sqrt{\sum_1^{11}((1 - 0) + (1 - 0) + (0 - 0,33) + (0 - 0) + (0 - 1) + \dots + (0 - 0))^2}$$

$$D_3 = 2,2846$$

- Jarak antara data 2 (X_{2i}) dengan bobot 4 (W_{4i}):

$$D_4 = \sqrt{\sum_1^{11}((X_{2,1} - W_{4,1}) + (X_{2,2} - W_{4,2}) + \dots + (X_{2,11} - W_{4,11}))^2}$$

$$D_4 = \sqrt{\sum_1^{11}((1 + 0) + (1 - 0,17) + (0 - 0) + (0 - 0,17) + (0 - 0) + \dots + (0 - 1))^2}$$

$$D_4 = 2,1224$$

2. Tentukan *neuron* pemenang (D_j minimum):

- D_1 merupakan *Distance* terdekat.

3. *Update* bobot pada D_j minimum:

- D_j minimum = $D_1 = W_{1i}$

- Cek apakah kelas yang dimiliki data sesuai dengan kelas yang dimiliki bobot. Data 2 dan bobot 1 memiliki kelas yang sama, yaitu Klasik maka update bobot menggunakan Persamaan (2.5). Jika kelas yang dimiliki data dan bobot tidak sama maka gunakan Persamaan (2.6).

- *Update* bobot W_{1i} menggunakan persamaan (2.5):

$$W_{1,1}(\text{baru}) = 0,8277 + 0,01 (1 - 0,8277) = 0,8294$$

$$W_{1,2}(\text{baru}) = 0,4946 + 0,01 (1 - 0,4946) = 0,4997$$

...

$$W_{1,11}(\text{baru}) = 0 + 0,01 (0 - 0) = 0$$

4. Ulangi proses 2 sampai 4 kepada seluruh data.

5. *Update* laju pembelajaran α setiap telah melalui 1 iterasi menggunakan Persamaan (2.7):

$$\alpha(\text{baru}) = 0,1 * 0,01$$

$$\alpha(\text{baru}) = 0,001$$

6. Periksa syarat berhenti (iterasi). Hasil perhitungan jarak pada iterasi 2 ditampilkan pada Tabel 4.8 dan hasil perbaikan bobot pada iterasi 2 akan ditampilkan pada Tabel 4.9.

Tabel 4.8 Hasil Perhitungan Jarak Iterasi 2 *SOM-LVQ*

Data	Jarak Iterasi 2 <i>SOM-LVQ</i>			
	d1	d2	d3	d4
2	0.560305	1.760596	2.284621	2.122382
3	0.986198	2.024772	2.493891	1.962903
4	1.518219	1.048666	2.284621	1.582563
5	0.983482	1.442086	2.054141	1.872033

Tabel 4.8 Hasil Perhitungan Jarak Iterasi 2 *SOM-LVQ* (Lanjutan)

Data	Jarak Iterasi 2 <i>SOM-LVQ</i>			
	d1	d2	d3	d4
7	1.994227	0.840358	2.284621	1.359099
8	1.897301	1.040108	2.284621	0.923573
9	2.151442	1.300826	2.493891	1.098272
10	1.986955	0.81555	1.794295	1.582563
12	1.722613	2.154992	0.756704	2.199524
13	2.228743	2.154992	0.741984	2.199524
14	1.991807	1.282181	1.491296	1.355694
15	2.442805	2.382653	0.94115	2.416176
17	2.155923	1.644462	2.287565	0.736763
18	2.235228	1.644462	2.287565	1.352095
19	1.897301	1.04013	2.287565	0.948612
20	1.406037	1.640802	2.287565	1.961271

Tabel 4.9 Hasil Perbaikan Bobot Iterasi 2 *SOM-LVQ*

	BOBOT AKHIR ITERASI 2 <i>SOM-LVQ</i>											Kelas
w1	0.831 041	0.504 749	0	0	0	0	0.175 861	- 0.024 62	1	0	0	Klasi k
w2	- 0.020 88	1.006 012	0	0	0.208 431	0	0	0.194 819	0	0	1	Muro ttal
w3	0	0	0.348 768	0	1	0.333 641	0	0	0	1	0	Rock
w4	0	0.161 548	0	0.182 363	0	0	0.817 637	0.335 324	0	0	1	Pop

7. Setelah pelatihan sudah mencapai batas iterasi yang ditentukan, dapat dilakukan *testing* atau proses klasifikasi vektor data input terhadap vektor bobot hasil pelatihan pada Tabel 4.9. *Testing* pada *SOM-LVQ* dilakukan dengan cara perhitungan jarak vektor data *input* dengan seluruh vektor bobot (sejumlah kelas). Jarak terdekat akan menentukan data tersebut tergolong dalam kelas yang dimiliki oleh bobot (*Vector Reference*). Tabel 4.10 merupakan hasil *Testing* data terhadap bobot hasil pelatihan pada Tabel 4.9.

Tabel 4.10 Tabel Perhitungan Akurasi SOM-LVQ

Da ta	FITUR											Jarak				W	Kela s
	x 1	x 2	x 3	x 4	x 5	x 6	x 7	x 8	x 9	x 1 0	x 1 1	d1	d2	d3	d4		
1	1	1	0	0	0	0	0	0	1	0	0	0.55 2589	1.76 7381	2.28 7565	2.12 5378	w 1	Klasi k
2	1	1	0	0	0	0	1	0	1	0	0	0.97 6541	2.03 0674	2.49 6589	1.97 0268	w 1	Klasi k
3	1	1	0	0	0	0	0	0	0	0	1	1.51 8339	1.06 0017	2.28 7565	1.58 6579	w 2	Mur ottal
4	0	1	0	0	0	0	0	0	1	0	0	0.98 3583	1.44 2869	2.05 7415	1.87 5428	w 1	Klasi k
5	0	1	0	0	0	0	0	1	0	0	1	2.00 4165	0.83 2005	2.28 7565	1.35 889	w 2	Mur ottal
6	0	1	0	0	0	0	1	0	0	0	1	1.90 1503	1.04 013	2.28 7565	0.93 9126	w 2	Mur ottal
7	0	1	0	0	0	0	1	1	0	0	1	2.15 9851	1.30 0858	2.49 6589	1.10 0594	w 4	Pop
8	0	1	0	0	1	0	0	0	0	0	1	1.99 1842	0.81 5481	1.79 8042	1.58 6579	w 2	Mur ottal
9	0	0	0	0	1	1	0	0	0	1	0	2.23 0904	2.16 2645	0.75 2113	2.20 0074	w 3	Rock
10	0	0	1	0	1	0	0	0	0	1	0	2.23 0904	2.16 2645	0.73 1724	2.20 0074	w 3	Rock
11	0	0	0	0	1	0	0	0	0	0	1	1.99 4225	1.29 5003	1.49 4308	1.35 6587	w 2	Mur ottal
12	0	0	1	0	1	1	0	0	0	1	0	2.44 4777	2.38 2653	0.93 1739	2.41 6677	w 3	Rock
13	0	0	0	0	0	0	1	1	0	0	1	2.16 2048	1.64 4462	2.28 7565	0.73 1029	w 4	Pop
14	0	0	0	1	0	0	0	1	0	0	1	2.24 1913	1.64 4462	2.28 7565	1.34 3485	w 4	Pop
15	0	1	0	0	0	0	1	0	0	0	1	1.90 1503	1.04 013	2.28 7565	0.93 9126	w 4	pop
16	0	1	0	0	0	0	0	1	1	0	0	1.42 0097	1.64 0802	2.28 7565	1.96 1271	w 1	Klasi k

Tabel 4.10 menyatakan bahwa akurasi yang dihasilkan dari pelatihan SOM-LVQ 2 iterasi dengan *learning rate* 0,1 dan *decrement* 0,1 sebesar 75%. Nilai tersebut didapatkan dari perhitungan jarak terdekat yang sesuai dengan kelasnya. Pada Tabel 4.10 terdapat 12 data yang diklasifikasikan dengan benar dari 16 data, maka 12/16 adalah 0,75 atau 75%.

4.4 Perancangan Uji Coba dan Evaluasi

Perancangan uji coba dan evaluasi dirancang agar pengujian untuk mendapatkan variabel-variabel terbaik sudah terkonsep. Pengujian yang dirancang meliputi algoritme yang digunakan, antara lain *SOM*, *LVQ* dan *SOM-LVQ*.

4.4.1 Perancangan Uji Coba *SOM*

Pengujian *SOM* dirancang agar mengetahui kombinasi terbaik dari pemilihan variabel. Kombinasi terbaik dari *learning rate* (α) dan *decrement learning rate* ($Dec \alpha$) berpengaruh terhadap klaster yang terbentuk dan juga nilai bobot yang dihasilkan, maka dari itu nilai dari variabel tersebut harus diuji. Iterasi yang terbaik ataupun iterasi yang sudah konvergen juga diuji agar mengetahui iterasi yang memberikan hasil maksimal pada pelatihan. Nilai yang dijadikan penentu untuk pemilihan variabel ataupun iterasi terbaik adalah persentase dari pengelompokan/pengklasteran data. Persentase digunakan untuk mempermudah dalam menentukan nilai terbaik. Pengujian pada *SOM* dirancang seperti Tabel 4.11, Tabel 4.12 dan Tabel 4.13.

Tabel 4.11 merupakan perancangan uji coba untuk menentukan *learning rate* terbaik pada pelatihan *SOM*. Rentang nilai *alpha/learning rate* dari 0,1 hingga 1 akan diuji persentase keakuratan dalam mengelompokkan data. Terdapat 4 variasi data yang digunakan untuk menguji *learning rate* terbaik dan juga nilai maksimum yang didapatkan.

Tabel 4.11 Uji Coba *Learning Rate SOM*

Learning rate	D1	D2	D3	D4	Maks
0.1					
0.2					
0.3					
0.4					
0.5					
0.6					
0.7					
0.8					
0.9					
1					

Tabel 4.12 merupakan perancangan uji coba untuk menentukan *decrement learning rate* terbaik pada pelatihan *SOM*. Rentang nilai *decrement* dari 0,1 hingga 1 akan diuji persentase keakuratan dalam mengelompokkan data. Terdapat 4 variasi data yang digunakan untuk menguji *decrement learning rate* terbaik dan juga nilai maksimum yang didapatkan.

Tabel 4.12 Uji Coba *Decrement SOM*

Decrement	D1	D2	D3	D4	Maks
0.1					
0.2					
0.3					
0.4					
0.5					
0.6					
0.7					
0.8					
0.9					
1					

Tabel 4.13 merupakan perancangan uji coba untuk menentukan iterasi terbaik pada pelatihan *SOM*. Rentang iterasi yang diuji adalah 1, 2, 4, 6, 15, 100, 1000 dengan tujuan melihat perubahan dari rentang iterasi kecil hingga besar dan juga batas konvergen. Terdapat 4 variasi data yang digunakan untuk menguji menentukan iterasi terbaik dan juga nilai maksimum yang didapatkan.

Tabel 4.13 Uji Coba Iterasi *SOM*

Iterasi	D1	D2	D3	D4	Maks
1					
2					
3					
7					
15					
100					
1000					

4.4.2 Perancangan Uji Coba *LVQ*

Pengujian *LVQ* dirancang agar mengetahui kombinasi terbaik dari pemilihan variabel. Kombinasi terbaik dari *learning rate* (α) dan *decrement learning rate* ($Dec \alpha$) berpengaruh terhadap nilai bobot yang dihasilkan dan juga hasil klasifikasi, maka dari itu nilai dari variabel tersebut harus diuji. Iterasi yang terbaik ataupun iterasi yang sudah konvergen juga diuji agar mengetahui iterasi yang memberikan hasil maksimal pada pelatihan. Pengujian variabel dan iterasi terbaik dari *LVQ* dibutuhkan untuk membandingkan serta analisis global terhadap hasil akhir pelatihan *SOM-LVQ*. Perancangan uji coba *LVQ* dirancang seperti Tabel 4.14, Tabel 4.15 dan Tabel 4.16.

Tabel 4.14 merupakan perancangan uji coba untuk menentukan *learning rate* terbaik pada pelatihan *LVQ*. Rentang nilai *alpha/learning rate* dari 0,1 hingga 1 akan diuji persentase keakuratan dalam klasifikasi data. Terdapat 4 variasi data

yang digunakan untuk menguji *learning rate* terbaik dan juga nilai maksimum yang didapatkan.

Tabel 4.14 Uji Coba *Learning Rate LVQ*

Learning rate	D1	D2	D3	D4	Maks
0.1					
0.2					
0.3					
0.4					
0.5					
0.6					
0.7					
0.8					
0.9					
1					

Tabel 4.15 merupakan perancangan uji coba untuk menentukan *decrement learning rate* terbaik pada pelatihan *LVQ*. Rentang nilai *decrement* dari 0,1 hingga 1 akan diuji persentase keakuratan dalam klasifikasi data. Terdapat 4 variasi data yang digunakan untuk menguji *decrement learning rate* terbaik dan juga nilai maksimum yang didapatkan.

Tabel 4.15 Uji Coba *Decrement LVQ*

Decrement	D1	D2	D3	D4	Maks
0.1					
0.2					
0.3					
0.4					
0.5					
0.6					
0.7					
0.8					
0.9					
1					

Tabel 4.16 merupakan perancangan uji coba untuk menentukan iterasi terbaik pada pelatihan *LVQ*. Rentang iterasi yang diuji adalah 1, 2, 4, 6, 15, 100, 1000 dengan tujuan melihat perubahan dari rentang iterasi kecil hingga besar dan juga batas konvergen. Persentase keakuratan klasifikasi digunakan untuk menentukan nilai terbaik. Terdapat 4 variasi data yang digunakan untuk menguji menentukan iterasi terbaik dan juga nilai maksimum yang didapatkan.

Tabel 4.16 Uji Coba Iterasi LVQ

Iterasi	D1	D2	D3	D4	Maks
1					
2					
3					
7					
15					
100					
1000					

4.4.3 Perancangan Uji Coba SOM-LVQ

Pengujian *SOM-LVQ* dirancang agar mengetahui kombinasi terbaik dari pemilihan variabel. Meskipun nilai bobot sudah dilatih melalui pelatihan *SOM*, kombinasi terbaik dari *learning rate* (α) dan *decrement learning rate* ($Dec \alpha$) juga perlu diketahui untuk memberikan hasil yang optimal, maka dari itu kombinasi dari variabel tersebut harus diuji. Iterasi yang terbaik ataupun iterasi yang sudah konvergen juga diuji agar mengetahui iterasi yang memberikan hasil maksimal pada pelatihan. Perancangan uji coba *SOM-LVQ* dirancang seperti Tabel 4.17, Tabel 4.18 dan Tabel 4.19.

Tabel 4.17 merupakan perancangan uji coba untuk menentukan *learning rate* terbaik pada pelatihan *SOM-LVQ*. Rentang nilai *alpha/learning rate* dari 0,1 hingga 1 akan diuji persentase keakuratan dalam klasifikasi data. Terdapat 4 variasi data yang digunakan untuk menguji *learning rate* terbaik dan juga nilai maksimum yang didapatkan.

Tabel 4.17 Uji Coba Learning Rate SOM-LVQ

Learning rate	D1	D2	D3	D4	Maks
0.1					
0.2					
0.3					
0.4					
0.5					
0.6					
0.7					
0.8					
0.9					
1					

Tabel 4.18 merupakan perancangan uji coba untuk menentukan *decrement learning rate* terbaik pada pelatihan *LVQ*. Rentang nilai *decrement* dari 0,1 hingga 1 akan diuji persentase keakuratan dalam klasifikasi data. Terdapat 4 variasi data yang digunakan untuk menguji *decrement learning rate* terbaik dan juga nilai maksimum yang didapatkan.

Tabel 4.18 Uji Coba *Decrement SOM-LVQ*

Decrement	D1	D2	D3	D4	Maks
0.1					
0.2					
0.3					
0.4					
0.5					
0.6					
0.7					
0.8					
0.9					
1					

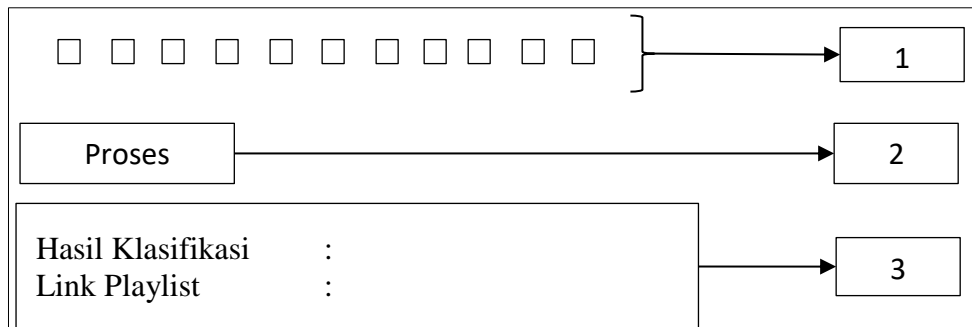
Tabel 4.19 merupakan perancangan uji coba untuk menentukan iterasi terbaik pada pelatihan *SOM-LVQ*. Rentang iterasi yang diuji adalah 1, 2, 4, 6, 15, 100, 1000 dengan tujuan melihat perubahan dari rentang iterasi kecil hingga besar dan juga batas konvergen. Persentase keakuratan klasifikasi digunakan untuk menentukan nilai terbaik. Terdapat 4 variasi data yang digunakan untuk menguji menentukan iterasi terbaik dan juga nilai maksimum yang didapatkan.

Tabel 4.19 Uji Coba Iterasi *LVQ*

Iterasi	D1	D2	D3	D4	Maks
1					
2					
3					
7					
10					
100					
1000					

4.5 Perancangan Antarmuka

Antarmuka dirancang untuk mempermudah pengguna memasukan data yang nantinya diproses menjadi suatu hasil klasifikasi jenis audio beserta *playlist*-nya. Antarmuka digunakan untuk menguji satu data inputan dari pengguna. Perancangan antarmuka ditampilkan pada Gambar 4.8.



Gambar 4.8 Perancangan Antarmuka

Perancangan antarmuka pada Gambar 4.6 terdiri dari 4 bagian yang memiliki fungsi tersendiri. Penjelasan dari perancangan tersebut antara lain:

1. Pada baris pertama yang ditandai dengan nomor 1 adalah 11 *input checkbox* dari pengguna berupa kondisi psikologi (emosi) yang terdiri dari konsentrasi, tenang, semangat, ceria, bebas, bertenaga, sendu, senang. Aktivitas yang dijadikan fitur yaitu kegiatan berfokus, berolahraga dan istirahat.
2. Pada baris kedua yang ditandai dengan nomor 2 adalah tombol untuk melakukan proses klasifikasi. Jika tombol tersebut ditekan maka proses pelatihan dan *testing* akan dijalankan sesuai dengan apa yang sudah diatur pada kode program.
3. Pada baris ketiga yang ditandai dengan nomor 3 adalah hasil klasifikasi. Hasil klasifikasi berupa label kelas. Terdapat 4 kelas yang dapat menjadi keluaran yaitu klasik, *murottal Al-Qur'an*, rock dan pop.

BAB 5 IMPLEMENTASI

5.1 Implementasi Data

Pada penelitian ini data diimplementasikan dalam bentuk dokumen CSV sesuai dengan perancangan pada deskripsi masalah. Kolom pertama merepresentasikan kelas dari data, kolom kedua hingga kesebelas merepresentasikan fitur dalam bentuk biner. Angka 1 merepresentasikan ya, angka 0 merepresentasikan tidak. Kelas 0, 1, 2 dan 3 merupakan klasik, Murottal, pop dan rock. Fitur *X1* hingga *X11* merupakan konsentrasi, tenang, semangat, ceria, bebas, bertenaga, sendu, senang, kegiatan berfokus, berolahraga dan istirahat. Contoh struktur data diimplementasikan seperti Tabel 5.1.

Tabel 5.1 Contoh Implementasi Data

Kelas	<i>X1</i>	<i>X2</i>	<i>X3</i>	<i>X4</i>	<i>X5</i>	<i>X6</i>	<i>X7</i>	<i>X8</i>	<i>X9</i>	<i>X10</i>	<i>X11</i>
0	1	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	1	0	0
0	1	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0	0
1	0	1	0	0	0	0	0	1	0	0	1
1	0	1	0	0	0	0	1	0	0	0	1
1	0	1	0	0	0	0	1	1	0	0	1
1	0	1	0	0	1	0	0	0	0	0	1
2	0	0	0	0	1	1	0	0	0	1	0
2	0	0	1	0	1	0	0	0	0	1	0
2	0	0	0	0	1	0	0	0	0	0	1
2	0	0	1	0	1	1	0	0	0	1	0
3	0	0	0	0	0	0	1	1	0	0	1
3	0	0	0	1	0	0	0	1	0	0	1
3	0	1	0	0	0	0	1	0	0	0	1
3	0	1	0	0	0	0	0	1	1	0	0

Dokumen CSV perlu diolah agar dapat diproses, maka diperlukan *source code library* yang berfungsi untuk membaca dan mengolah dokumen tersebut menjadi data yang dapat digunakan. Implementasi dari *library* tersebut ditampilkan pada Tabel 5.2.

Tabel 5.2 *Source Code Dataset*

No	Source Code
1	<code>package lvq;</code>
2	
3	<code>import java.io.IOException;</code>
4	<code>import java.net.URI;</code>
5	<code>import java.net.URISyntaxException;</code>
6	<code>import java.nio.charset.Charset;</code>

Tabel 5.2 Source Code Dataset (Lanjutan)

No	Source Code
7	import java.nio.file.Files;
8	import java.nio.file.Paths;
9	import java.util.*;
10	
11	class Data {
12	
13	int target;
14	double[] input = new double[11];
15	}
16	public class Dataset {
17	private static List<String> readFile(String fileName) {
18	List<String> lines = null;
19	try {
20	URI uri =
	Dataset.class.getResource(fileName).toURI();
21	lines = Files.readAllLines(Paths.get(uri),
	Charset.defaultCharset());
22	} catch (URISyntaxException IOException e) {
23	System.out.println(e);
24	}
25	return lines;
26	}
27	
28	public static Data[] dataLatih() {
29	ArrayList<Data> dataset = new ArrayList<>();
30	List<String> file = readFile("data_latih.csv");
31	for (String line : file) {
32	String[] array = line.split(",");
33	Data data = new Data();
34	data.target = Integer.parseInt(array[0]);
35	for (int i = 0; i < data.input.length; i++) {
36	data.input[i] = Double.parseDouble(array[1
	+ i]);
37	}
38	dataset.add(data);
39	}
40	return dataset.toArray(new Data[dataset.size()]);
41	}
42	
43	public static Data[] dataBobot() {
44	ArrayList<Data> dataset = new ArrayList<>();
45	List<String> file = readFile("data_bobot.csv");

Tabel 5.2 Source Code Dataset (Lanjutan)

No	Source Code
46	for (String line : file) {
47	String[] array = line.split(",");
48	Data data = new Data();
49	data.target = Integer.parseInt(array[0]);
50	for (int i = 0; i < data.input.length; i++) {
51	data.input[i] = Double.parseDouble(array[1
	+ i]);
52	}
53	dataset.add(data);
54	}
55	return dataset.toArray(new Data[dataset.size()]);
56	}
57	
58	public static String[] dataLabel() {
59	List<String> file = readFile("label_target.csv");
60	return file.toArray(new String[file.size()]);
61	}
62	}

Penjelasan:

Baris 3-9: *Import* semua fungsi yang dibutuhkan.

Baris 11-15: Deklarasi kelas *Data* untuk dijadikan objek dengan isi *array* bertipe *double* dan panjang 11 data.

Baris 16: Deklarasi kelas *Dataset*. Kelas ini berfungsi menampung logika untuk membaca *file* dan juga membentuk *array list*.

Baris 17-26: Deklarasi fungsi *readfile* untuk membaca *file*.

Baris 28-41: Deklarasi fungsi *dataLatih* bertipe *objek* dari kelas *Data*. Fungsi ini berguna untuk membaca *file* dari *data_latih.csv* dan dijadikan *array list*. Perulangan dilakukan sebanyak baris yang ada pada *data_latih.csv*. Setiap kali perulangan dilakukan proses pemisahan data berdasarkan simbol “,” (koma) pada setiap barisnya untuk dimasukkan kedalam array. Indeks ke-0 merupakan nilai kelas dari tiap baris/vektor.

Baris 43-56: Deklarasi fungsi *dataBobot* bertipe *objek* dari kelas *Data*. Fungsi ini berguna untuk membaca *file* dari *data_bobot.csv* dan dijadikan *array list*. Perulangan dilakukan sebanyak baris yang ada pada *data_latih.csv*. Setiap kali perulangan dilakukan proses pemisahan data berdasarkan simbol “,” (koma) pada setiap barisnya untuk dimasukkan kedalam array. Indeks ke-0 merupakan nilai kelas dari tiap baris/vektor.

Baris 58-62: Deklarasi fungsi *dataLabel* bertipe *array string* yang berguna untuk membaca *file label_target.csv* sebagai nama dari kelas tiap data sesuai indeksinya.

5.2 Implementasi Algoritme

Algoritme *Self Organizing Maps (SOM)* dan *Learning Vector Quantization (LVQ)* pada penelitian ini diimplementasikan menggunakan bahasa *java*. Terdapat 2 kelas untuk implementasi kedua algoritme tersebut. Kelas SOM untuk algoritme *Self Organizing Maps* dan kelas LVQ untuk algoritme *Learning Vector Quantization*. Algoritme beserta fungsi pendukung yang sudah dirancang dijelaskan pada tabel *source code* tiap subbab implementasi algoritme berikut.

5.2.1 Self Organizing Maps

Algoritme *Self Organizing Maps (SOM)* diimplementasikan kedalam kelas SOM. Pada kelas SOM terdapat beberapa fungsi yang mendukung dan menyusun algoritme *Self Organizing Maps* sesuai dengan bab perancangan. Implementasi algoritme pelatihan SOM dalam kode program dipaparkan pada Tabel 5.3.

Tabel 5.3 Source Code Pelatihan SOM

No	Source Code
1	<code>package lvq;</code>
2	<code>import java.text.DecimalFormat;</code>
3	
4	<code>public class SOM {</code>
5	
6	<code> public Data_SOM[] trainingInput;</code>
7	<code> public Data_SOM[] trainingWeight;</code>
8	<code> public String[] labelTarget;</code>
9	
10	<code> private double learningRate;</code>
11	<code> private double decrementRate;</code>
12	<code> private int iteration;</code>
13	
14	<code> public static void main(String[] args) {</code>
15	<code> SOM som = new SOM();</code>
16	<code> som.initialize(0.1, 0.1, 2);</code>
17	<code> som.train();</code>
18	
19	<code> Data_SOM inputTest = som.trainingInput[1];</code>
20	<code> int target = inputTest.target;</code>
21	<code> int hasil = som.test(inputTest.input);</code>
22	
23	<code> System.out.println("Keluaran sistem : " +</code>
	<code> som.labelTarget[hasil]);</code>
24	<code> System.out.println("Target input : " +</code>
	<code> som.labelTarget[target]);</code>
25	
26	<code> }</code>

Tabel 5.3 *Source Code* Pelatihan SOM (Lanjutan)

No	Source Code
27	
28	public void initialize(double learnRate, double
29	decRate, int iter) {
30	learningRate = learnRate;
31	decrementRate = decRate;
32	iteration = iter;
33	trainingInput = Dataset_SOM.dataLatih();
34	trainingWeight = Dataset_SOM.dataBobot();
35	labelTarget = Dataset_SOM.dataLabel();
36	}
37	public void train() {
38	for (int i = 0; i < iteration; i++) {
39	double[] currentDistance = new
40	double[trainingWeight.length];
41	for (int j = 0; j < trainingInput.length; j++)
42	{
43	for (int k = 0; k < currentDistance.length;
44	k++) {
45	double[] arr1 = trainingInput[j].input;
46	double[] arr2 =
47	trainingWeight[k].input;
48	currentDistance[k] =
49	euclidDistance(arr1, arr2);
50	}
51	int weightTarget =
52	minimumDistance(currentDistance);
	updateWeight(weightTarget, j);
	}
	learningRate = decrementRate * learningRate;
	}
	}

Penjelasan:

Baris 2: Menggunakan */import* fungsi dari *java* yang bernama *DecimalFormat* untuk mengola format desimal.

Baris 4: Deklarasi kelas *SOM*.

Baris 6-7: Inisialisasi variabel *array trainingInput* dan *trainingWeight* dari kelas *Data_SOM*.

Baris 8: Inisialisasi variabel *array String labelTarget*.

Baris 10-11: Inisialisasi variabel *double learningRate* dan *decrementRate*.

Baris 12: Inisialisasi variabel *integer iteration*.

Baris 14: Deklarasi fungsi *Main* pada kelas *SOM* untuk menjalankan fungsi-fungsi didalamnya.

Baris 15: Membuat objek dari kelas *SOM* yang bernama *som*.

Baris 16: Memanggil fungsi *initialize* dari kelas *SOM* dengan parameter *learningRate*, *decrementRate* dan *iteration*.

Baris 17: Memanggil fungsi *train* dari kelas *SOM*.

Baris 28-35: Deklarasi fungsi *initialize* yang berparameter *learnRate*, *decRate* dan *iter*. Fungsi tersebut berguna untuk inisialisasi *learning rate*, *decrement*, iterasi, data latih dan data bobot pada proses pelatihan *SOM*.

Baris 37-52: Deklarasi fungsi *train* untuk melakukan proses pelatihan sebanyak iterasi maksimum. Setiap 1 iterasi dilakukan proses perhitungan jarak antara vektor data latih dengan vektor data bobot sebanyak vektor data latih yang ada. Setiap 1 vektor data latih dilakukan proses perhitungan jarak terhadap semua vektor data bobot menggunakan fungsi *euclidDistance*, lalu ditentukan nilai jarak dengan bobot terkecil menggunakan fungsi *minimumDistance* untuk mengetahui vektor bobot yang akan diperbaiki menggunakan fungsi *updateWeight*. Setiap akhir iterasi dilakukan update pada *learningRate* dengan cara mengalikan nilai *learningRate* dengan *decrementRate*.

Salah satu proses yang terdapat pada pelatihan *SOM* adalah perhitungan jarak menggunakan *euclidean distance*. Pada Tabel 5.3 tepatnya fungsi *train* baris ke-45, fungsi *euclidDistance* digunakan/dipanggil. Fungsi *euclidDistance* pada kelas *SOM* merepresentasikan proses perhitungan jarak menggunakan *euclidean distance*. Tabel 5.4 adalah implementasi kode program untuk proses perhitungan jarak dengan nama fungsi *euclidDistance*.

Tabel 5.4 Source Code Euclidean Distance

No	Source Code
1	<code>private double euclidDistance(double[] arr1, double[] arr2)</code>
2	<code>{</code>
3	<code> double dist = 0;</code>
4	<code> for (int i = 0; i < arr1.length; i++) {</code>
5	<code> dist += Math.pow(arr1[i] - arr2[i], 2);</code>
6	<code> }</code>
7	<code> dist = Math.sqrt(dist);</code>
8	<code> return dist;</code>
9	<code>}</code>

Penjelasan:

Baris 1: Deklarasi fungsi *euclideanDistance* yang memiliki parameter 2 *array* bertipe *double* sebagai *input* data yang akan dilakukan proses perhitungan jarak.

Baris 3: Inisialisasi variabel *double dist* yang berfungsi menyimpan nilai jarak.

Baris 4-9: Perulangan sebanyak panjang indeks *array*. Setiap indeks pada 2 *array* dilakukan proses perhitungan jarak menggunakan Persamaan (2.1). Setelah proses perhitungan selesai, simpan nilai variabel *dist* yang telah menyimpan nilai hasil perhitungan jarak.

Pada Tabel 5.3 tepatnya fungsi *train* baris ke-47, fungsi *minimumDistance* digunakan/dipanggil. Fungsi *minimumDistance* pada kelas *SOM* merepresentasikan proses pemilihan jarak minimum dengan cara mengganti nilai variabel dengan nilai terkecil melalui proses perbandingan nilai. Tabel 5.5 adalah implementasi kode program untuk proses pemilihan jarak minimum.

Tabel 5.5 *Source Code Minimum Distance*

No	Source Code
1	<code>private int minimumDistance(double[] distances) {</code>
2	<code> int minIndex = -1;</code>
3	<code> double min = Double.MAX_VALUE;</code>
4	<code> for (int i = 0; i < distances.length; i++) {</code>
5	<code> if (distances[i] < min) {</code>
6	<code> min = distances[i];</code>
7	<code> minIndex = i;</code>
8	<code> }</code>
9	<code> }</code>
10	<code> return trainingWeight[minIndex].target;</code>
11	<code>}</code>

Penjelasan:

Baris 1: Deklarasi fungsi *minimumDistance* dengan parameter *array* yang berisi nilai jarak 1 vektor data dengan 4 vektor bobot.

Baris 2-3: Inisialisasi variabel yang mewakili nilai besar yaitu *minIndex* dan nilai kecil yaitu *min*. Kedua variabel tersebut berfungsi menjadi pembanding untuk mencari jarak yang memiliki nilai terkecil.

Baris 4-11: Lakukan proses perulangan sebanyak indeks *array distances*. Setiap 1 proses bandingkan nilai distance indeks ke-*i* menggunakan logika *if*, yaitu jika nilai jarak pada indeks ke-*i* lebih kecil dari nilai *min* maka simpan nilai vektor *distances* indeks ke-*i* menjadi nilai terkecil dan juga indeks terkecil, jika tidak maka abaikan. Jika semua proses sudah selesai, simpan indeks vektor terkecil untuk digunakan sebagai indeks dari vektor bobot.

Tabel 5.3 tepatnya pada fungsi *train* baris ke-48, fungsi *updateWeight* digunakan/dipanggil. Fungsi *updateWeight* pada kelas *SOM* merepresentasikan proses perbaikan bobot/*update* bobot. Perbaikan bobot atau *update* bobot pada pelatihan *SOM* berfungsi untuk memperbaiki nilai bobot agar lebih baik dalam mengklaster data. Tabel 5.6 adalah implementasi kode program untuk proses *update* bobot pada pelatihan *SOM*.

Tabel 5.6 *Source Code Update Bobot SOM*

No	Source Code
1	<code>private void updateWeight(int cj, int t) {</code>
2	
3	<code> double[] data = trainingInput[t].input;</code>
4	<code> double[] weight = trainingWeight[cj].input;</code>
5	<code> for (int i = 0; i < weight.length; i++) {</code>
6	<code> weight[i] = weight[i] + learningRate * (data[i]</code>
7	<code> - weight[i]);</code>
8	<code> }</code>
9	<code> trainingWeight[cj].input = weight;</code>
10	<code>}</code>

Penjelasan:

Baris 1: Deklarasi fungsi *updateWeight* dengan 2 parameter bertipe *integer* yaitu *cj* dan *t*.

Baris 3: Inisialisasi array data yang berisi nilai dari vektor data latih ke-*t*.

Baris 4: Inisialisasi *array weight* yang berisi nilai dari vektor data bobot ke-*cj*.

Baris 5-9: Perulangan sebanyak indeks dari *array weight*. Setiap 1 proses lakukan perhitungan menggunakan Persamaan (2.2) sesuai dengan indeks ke-*i* pada masing-masing array. Setelah semua proses telah selesai, perbarui nilai bobot dengan nilai yang telah diperbaiki.

5.2.2 Learning Vector Quantization

Algoritme *Learning Vector Quantization (LVQ)* diimplementasikan kedalam kelas LVQ. Pada kelas LVQ terdapat beberapa fungsi yang mendukung dan menyusun algoritme *Learning Vector Quantization* sesuai dengan bab perancangan. Implementasi algoritme pelatihan LVQ dalam kode program dipaparkan pada Tabel 5.7.

Tabel 5.7 Source Code Pelatihan LVQ

No	Source Code
1	<code>package lvq;</code>
2	<code>import java.text.DecimalFormat;</code>
3	
4	<code>public class LVQ {</code>
5	
6	<code> public Data[] trainingInput;</code>
7	<code> public Data[] trainingWeight;</code>
8	<code> public String[] labelTarget;</code>
9	
10	<code> private double learningRate;</code>
11	<code> private double decrementRate;</code>
12	<code> private int iteration;</code>
13	
14	<code> public static void main(String[] args) {</code>
15	<code> LVQ lvq = new LVQ();</code>
16	<code> lvq.initialize(0.1, 0.1, 1);</code>
17	<code> lvq.train();</code>
18	
19	<code> Data inputTest = lvq.trainingInput[4];</code>
20	<code> int target = inputTest.target;</code>
21	<code> int hasil = lvq.test(inputTest.input);</code>
22	
23	<code> System.out.println("Keluaran sistem : " +</code>
	<code>lvq.labelTarget[hasil]);</code>
24	<code> System.out.println("Target input : " +</code>
	<code>lvq.labelTarget[target]);</code>

Tabel 5.7 Source Code Pelatihan LVQ (Lanjutan)

No	Source Code
25	}
26	
27	public void initialize(double learnRate, double decRate,
	int iter) {
28	learningRate = learnRate;
29	decrementRate = decRate;
30	iteration = iter;
31	trainingInput = Dataset.dataLatih();
32	trainingWeight = Dataset.dataBobot();
33	labelTarget = Dataset.dataLabel();
34	}
35	
36	public void train() {
37	for (int i = 0; i < iteration; i++) {
38	double[] currentDistance = new
	double[trainingWeight.length];
39	for (int j = 0; j < trainingInput.length; j++) {
40	for (int k = 0; k < currentDistance.length;
	k++) {
41	double[] arr1 = trainingInput[j].input;
42	double[] arr2 = trainingWeight[k].input;
43	currentDistance[k] =
	euclidDistance(arr1, arr2);
44	}
45	int weightTarget =
	minimumDistance(currentDistance);
46	updateWeight(weightTarget, j);
47	}
48	learningRate = decrementRate * learningRate;
49	}
50	}

Penjelasan:

Baris 2: Menggunakan/import fungsi dari *java* yang bernama *DecimalFormat* untuk mengola format *desimal*.

Baris 4: Deklarasi kelas *LVQ*.

Baris 6-7: Inisialisasi variabel *array trainingInput* dan *trainingWeight* dari kelas *Data*.

Baris 8: Inisialisasi variabel *array String labelTarget*.

Baris 10-11: Inisialisasi variabel *double learningRate* dan *decrementRate*.

Baris 12: Inisialisasi variabel *integer iteration*.

Baris 14: Deklarasi fungsi *Main* pada kelas *LVQ* untuk menjalankan fungsi-fungsi didalamnya.

Baris 15: Membuat objek dari kelas *LVQ* yang bernama *lvq*.

Baris 16: Memanggil fungsi *initialize* dari kelas *LVQ* dengan parameter *learningRate*, *decrementRate* dan *iteration*.

Baris 17: Memanggil fungsi *train* dari kelas *LVQ*.

Baris 27-34: Deklarasi fungsi *initialize* yang berparameter *learnRate*, *decRate* dan *iter*. Fungsi tersebut berguna untuk inisialisasi *learning rate*, *decrement*, iterasi, data latih dan data bobot pada proses pelatihan *LVQ*.

Baris 36-50: Deklarasi fungsi *train* untuk melakukan proses pelatihan sebanyak iterasi maksimum. Setiap 1 iterasi dilakukan proses perhitungan jarak antara vektor data latih dengan vektor data bobot sebanyak vektor data latih yang ada. Setiap 1 vektor data latih dilakukan proses perhitungan jarak terhadap semua vektor data bobot menggunakan fungsi *euclidDistance*, lalu ditentukan nilai jarak dengan bobot terkecil menggunakan fungsi *minimumDistance* untuk mengetahui vektor bobot yang akan diperbaiki menggunakan fungsi *updateWeight*. Setiap akhir iterasi dilakukan update pada *learningRate* dengan cara mengalikan nilai *learningRate* dengan *decrementRate*.

Salah satu proses yang terdapat pada pelatihan *LVQ* adalah perhitungan jarak menggunakan *euclidean distance*. Pada Tabel 5.7 tepatnya fungsi *train* baris ke-43, fungsi *euclidDistance* digunakan/dipanggil. Fungsi *euclidDistance* pada kelas *LVQ* merepresentasikan proses perhitungan jarak menggunakan *euclidean distance*. Tabel 5.4 adalah implementasi kode program untuk proses perhitungan jarak dengan nama fungsi *euclidDistance* yang juga digunakan pada kelas *LVQ*.

Pada Tabel 5.7 tepatnya fungsi *train* baris ke-45, fungsi *minimumDistance* digunakan/dipanggil. Fungsi *minimumDistance* pada kelas *LVQ* merepresentasikan proses pemilihan jarak minimum dengan cara mengganti nilai variabel dengan nilai terkecil melalui proses perbandingan nilai. Tabel 5.5 adalah implementasi kode program untuk proses pemilihan jarak minimum yang juga digunakan pada kelas *LVQ*.

Pada Tabel 5.7 tepatnya fungsi *train* baris ke-46, fungsi *updateWeight* digunakan/dipanggil. Fungsi *updateWeight* pada kelas *LVQ* merepresentasikan proses perbaikan bobot/*update* bobot. Perbaikan bobot atau *update* bobot pada pelatihan *LVQ* berfungsi untuk memperbaiki nilai bobot agar lebih baik dalam klasifikasi data. Tabel 5.8 adalah implementasi kode program untuk proses *update* bobot pada pelatihan *LVQ*.

Tabel 5.8 Source Code Update Bobot LVQ

No	Source Code
1	<code>private void updateWeight(int cj, int t) {</code>
2	<code> double alpha;</code>
3	<code> if (cj == trainingInput[t].target) {</code>
4	<code> alpha = learningRate;</code>
5	<code> } else {</code>
6	<code> alpha = -learningRate;</code>
7	<code> }</code>
8	<code> double[] data = trainingInput[t].input;</code>
9	<code> double[] weight = trainingWeight[cj].input;</code>
10	<code> for (int i = 0; i < weight.length; i++) {</code>
11	<code> weight[i] = weight[i] + alpha * (data[i] -</code>
12	<code> weight[i]);</code>
13	<code> }</code>
14	<code> trainingWeight[cj].input = weight;</code>
15	<code>}</code>

Penjelasan:

Baris 1: Deklarasi fungsi *updateWeight* dengan 2 parameter bertipe *integer* yaitu *cj* dan *t*.

Baris 2: Inisialisasi variabel *double* bernama *alpha*.

Baris 4-6: Operasi logika *if*, jika nilai dari parameter *input cj* bernilai sama dengan target dari vektor data ke-*t* maka nilai variabel *alpha = learningRate* (positif) dan jika tidak maka nilai variabel *alpha = -learningRate* (negatif).

Baris 7: Inisialisasi *array data* yang berisi nilai dari vektor data latih ke-*t*.

Baris 8: Inisialisasi *array weight* yang berisi nilai dari vektor data bobot ke-*cj*.

Baris 9-13: Perulangan sebanyak indeks dari *array weight*. Setiap 1 proses lakukan perhitungan menggunakan Persamaan (2.5) sesuai dengan indeks ke-*i* pada masing-masing *array*. Setelah semua proses telah selesai, perbarui nilai bobot dengan nilai yang telah diperbaiki.

Algoritme *Learning Vector Quantization* pada penelitian ini digunakan untuk klasifikasi data. Setelah semua proses pelatihan selesai, dilakukan proses *testing* untuk mengetahui suatu data *input* masuk dalam kelas apa. Proses *testing* adalah proses perhitungan jarak 1 vektor data *input* dengan seluruh vektor data bobot (perwakilan kelas) menggunakan *euclidean distance*. Nilai jarak vektor data *input* dengan salah satu vektor data bobot yang minimum menandakan bahwa data *input* tersebut adalah bagian dari kelas bobot tersebut. Fungsi test pada kelas LVQ merepresentasikan proses *testing*. Tabel 5.9 adalah implementasi kode program untuk proses *test* pada kelas LVQ.

Tabel 5.9 Source Code Test

No	Source Code
1	<code>public int test(double[] input) {</code>
2	<code> double[] dist = new double[trainingWeight.length];</code>
3	<code> for (int i = 0; i < trainingWeight.length; i++) {</code>
4	<code> dist[i] = euclidDistance(input,</code>
5	<code> trainingWeight[i].input);</code>

Tabel 5.9 Source Code Test (Lanjutan)

No	Source Code
6	}
7	int result = minimumDistance(dist);
8	return result;
9	}

Penjelasan:

Baris 1: Deklarasi fungsi *test* yang memiliki parameter *array input* bertipe *double*.

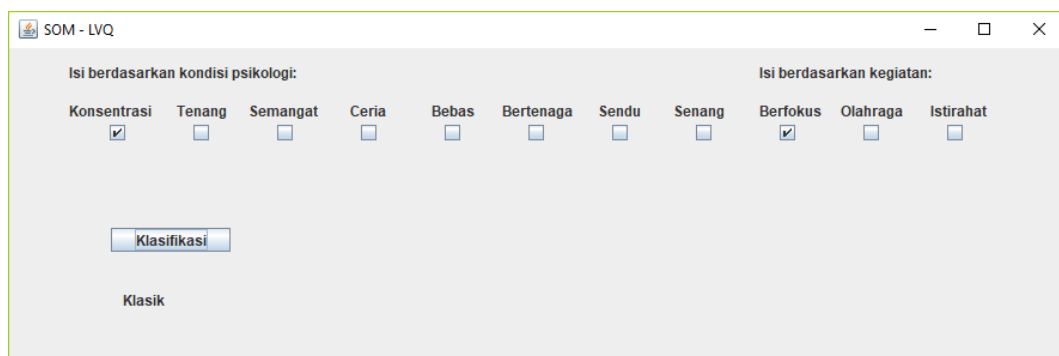
Baris 2: Inisialisasi *array* bernama *dist* bertipe *double* yang memiliki panjang sebanyak panjang dari *trainingWeight*.

Baris 3-6: Perulangan sebanyak panjang dari *trainingWeight*. Setiap 1 proses dilakukan perhitungan jarak menggunakan fungsi *euclidDistance* seperti pada Tabel 5.4. Setiap nilai jarak dari vektor data dengan vektor bobot ke-*i* disimpan dalam variabel *dist[i]*.

Baris 7-9: Inisialisasi variabel *result* yang memanggil fungsi *minimumDistance* seperti pada Tabel 5.5 untuk proses pemilihan nilai minimum dari variabel *dist*. Setelah proses pemilihan nilai minimum selesai, kembalikan/simpan nilai *result*.

5.3 Implementasi Antarmuka

Implementasi antarmuka diimplementasikan seperti perancangan antarmuka yang telah dibuat. Antarmuka atau *Graphical User Interface (GUI)* dibuat dengan bahasa *java* dan IDE NetBeans. Antarmuka berfungsi untuk memasukkan 1 data uji terhadap data yang sudah ditentukan. Data tersebut diproses dengan data latih dan bobot yang sudah ditentukan.



Gambar 5.1 Implementasi Antarmuka

BAB 6 PENGUJIAN DAN ANALISIS

6.1 Skenario Pengujian

Pengujian pada penelitian ini meliputi pemilihan variabel terbaik dan juga iterasi terbaik terhadap algoritme *SOM*, *LVQ* dan *SOM-LVQ*. Perancangan untuk pengujian telah dirancang pada subbab perancangan uji coba. Data bobot yang digunakan adalah data pertama dari masing-masing kelas pada data latih (Azmi, 2014). Tabel 6.1 adalah bobot yang digunakan untuk pengujian. Kelas 0, 1, 2 dan 3 merupakan klasik, Murottal, pop dan rock. Fitur *X1* hingga *X11* merupakan konsentrasi, tenang, semangat, ceria, bebas, bertenaga, sendu, senang, kegiatan berfokus, berolahraga dan istirahat. Angka 1 merepresentasikan ya, angka 0 merepresentasikan tidak.

Tabel 6.1 Data Bobot Pengujian

Kelas	<i>X1</i>	<i>X2</i>	<i>X3</i>	<i>X4</i>	<i>X5</i>	<i>X6</i>	<i>X7</i>	<i>X8</i>	<i>X9</i>	<i>X10</i>	<i>X11</i>
0	1	0	0	0	0	0	0	0	1	0	0
1	0	1	0	0	0	0	0	0	0	0	1
2	0	0	0	0	1	0	0	0	0	1	0
3	0	0	0	0	0	0	1	0	0	0	1

6.2 Pengujian SOM

Pengujian pada *SOM* dilakukan untuk menentukan semua komponen terbaik sesuai dengan apa yang sudah dirancang agar dapat memberikan bobot terbaik pada pelatihan selanjutnya. Pengujian yang dilakukan antara lain pengujian *Learning Rate*, *Decrement Learning Rate* dan Iterasi.

6.2.1 Pengujian *Learning Rate* Pada *SOM*

Tabel 6.2 merupakan pengujian untuk menentukan *learning rate* terbaik pada pelatihan *SOM*. *Decrement learning rate* awal bernilai 0,5 (Azmi, 2014). Iterasi yang digunakan adalah 100 iterasi. Unsur-unsur yang digunakan untuk pengujian ini antara lain:

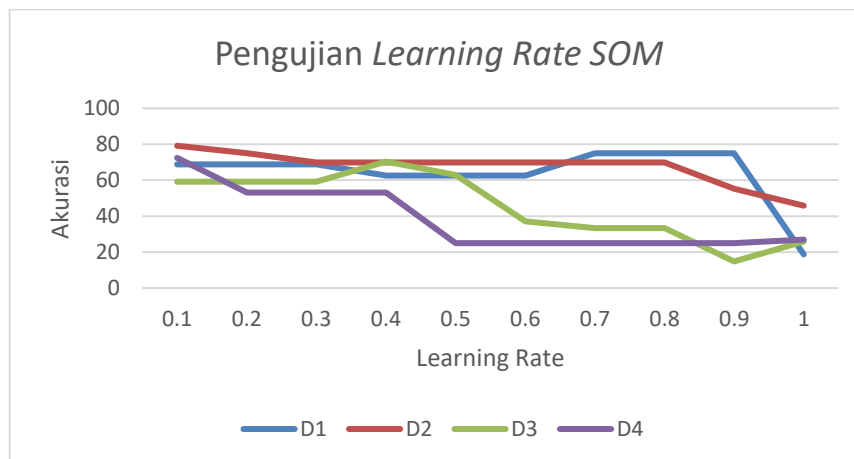
1. Rentang nilai *alpha/learning rate* dari 0,1 hingga 1.
2. *Dataset* awal yang menggunakan jumlah kasus (D1). Data terdapat pada Lampiran B.1.
3. *Dataset* awal utuh tanpa adanya jumlah kasus (D2). Data terdapat pada Lampiran B.2.
4. *Dataset* lanjut yang menggunakan jumlah kasus (D3). Data terdapat pada Lampiran B.3.
5. *Dataset* lanjut utuh tanpa adanya jumlah kasus (D4). Data terdapat pada Lampiran B.4.
6. Nilai hasil berupa persentase akurasi dari masing-masing dataset yang digunakan.
7. Nilai maksimum yang didapatkan (Maks).

Tabel 6.2 menampilkan hasil pengujian *learning rate* pada pelatihan SOM dari keseluruhan data yang digunakan. Nilai *learning rate* 0,1 pada *dataset* 2 (D2) mendapatkan nilai maksimum dari keseluruhan nilai akurasi yang didapat yaitu 79,166%. *Learning rate* 0,1 adalah variabel terbaik pada pengujian SOM.

Tabel 6.2 Pengujian *Learning Rate* SOM

Learning rate	D1	D2	D3	D4	Maks
0.1	68.75	79.166	59.259	72.436	79.166
0.2	68.75	75	59.259	53.205	75
0.3	68.75	69.792	59.259	53.205	69.792
0.4	62.5	69.792	70.37	53.205	70.37
0.5	62.5	69.792	62.963	25	69.792
0.6	62.5	69.792	37.037	25	69.792
0.7	75	69.792	33.33	25	75
0.8	75	69.792	33.33	25	75
0.9	75	55.208	14.815	25	75
1	18.75	45.833	25.925	26.923	45.833
Nilai Maks					79.166

Gambar 6.1 merupakan grafik pengujian *learning rate* pada pelatihan SOM. Grafik tersebut menunjukkan bahwa pada *dataset* 2 (D2) dengan nilai *learning rate* 0,1 meraih nilai maksimum, namun dari 0,2 hingga 1 terus mengalami penurunan nilai. *Dataset* 2 (D2) memberikan nilai maksimum pada pengujian daripada *dataset* lain, seperti yang sudah ditampilkan pada Tabel 6.2 dan juga Gambar 6.1.



Gambar 6.1 Grafik Pengujian *Learning Rate* SOM

6.2.2 Pengujian *Decrement Learning Rate* Pada SOM

Tabel 6.3 merupakan pengujian untuk menentukan *decrement learning rate* terbaik pada pelatihan SOM. *Learning rate* yang digunakan adalah 0,1 dikarenakan pada pengujian *learning rate* pelatihan SOM nilai 0,1 yang

memperoleh nilai maksimum. Iterasi yang digunakan adalah 100 iterasi. Unsur-unsur yang digunakan untuk pengujian ini antara lain:

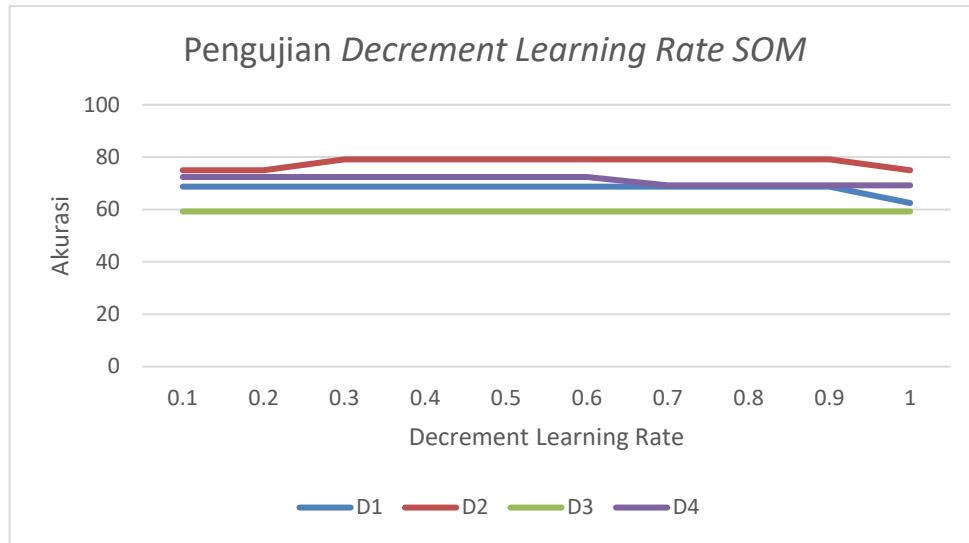
1. Rentang nilai *decrement learning rate* dari 0,1 hingga 1.
2. *Dataset* awal yang menggunakan jumlah kasus (D1). Data terdapat pada Lampiran B.1.
3. *Dataset* awal utuh tanpa adanya jumlah kasus (D2). Data terdapat pada Lampiran B.2.
4. *Dataset* lanjut yang menggunakan jumlah kasus (D3). Data terdapat pada Lampiran B.3.
5. *Dataset* lanjut utuh tanpa adanya jumlah kasus (D4). Data terdapat pada Lampiran B.4.
6. Nilai hasil berupa persentase akurasi dari masing-masing dataset yang digunakan.
7. Nilai maksimum yang didapatkan (Maks).

Tabel 6.3 menampilkan hasil pengujian *decrement learning rate* pada pelatihan *SOM* dari keseluruhan data yang digunakan. Nilai *decrement learning rate* 0,3 hingga 0,9 pada *dataset* 2 (D2) mendapatkan nilai maksimum dari keseluruhan nilai akurasi yang didapat yaitu 79,167%. Nilai *decrement learning rate* 0,3 hingga 0,9 adalah variabel terbaik.

Tabel 6.3 Pengujian *Decrement SOM*

Decrement	D1	D2	D3	D4	Maks
0.1	68.75	75	59.259	72.436	75
0.2	68.75	75	59.259	72.436	75
0.3	68.75	79.167	59.259	72.436	79.167
0.4	68.75	79.167	59.259	72.436	79.167
0.5	68.75	79.167	59.259	72.436	79.167
0.6	68.75	79.167	59.259	72.436	79.167
0.7	68.75	79.167	59.259	69.231	79.167
0.8	68.75	79.167	59.259	69.231	79.167
0.9	68.75	79.167	59.259	69.231	79.167
1	62.5	75	59.259	69.231	75
Nilai Maks					79.167

Gambar 6.2 merupakan grafik pengujian *decrement learning rate* pada pelatihan *SOM*. Grafik tersebut menunjukkan bahwa nilai *decrement* dari 0,3 hingga 0,9 pada *dataset* 2 (D2) menghasilkan nilai yang sama dan juga mencapai nilai maksimum, namun dari 0,9 menuju 1 mengalami penurunan. Seluruh *dataset* bernilai stabil, namun rata-rata nilai tertinggi dicapai dengan menggunakan *dataset* 2 (D2).



Gambar 6.2 Grafik Pengujian *Decrement SOM*

6.2.3 Pengujian Iterasi Pada *SOM*

Tabel 6.4 merupakan pengujian untuk menentukan iterasi terbaik pada pelatihan *SOM*. Unsur-unsur yang digunakan untuk pengujian ini antara lain:

1. Rentang iterasi yang diuji.
2. Nilai *learning rate* = 0,1 dan *decrement* = 0,3 sesuai dengan hasil terbaik pengujian sebelumnya.
3. *Dataset* awal yang menggunakan jumlah kasus (D1). Data terdapat pada Lampiran B.1.
4. *Dataset* awal utuh tanpa adanya jumlah kasus (D2). Data terdapat pada Lampiran B.2.
5. *Dataset* lanjut yang menggunakan jumlah kasus (D3). Data terdapat pada Lampiran B.3.
6. *Dataset* lanjut utuh tanpa adanya jumlah kasus (D4). Data terdapat pada Lampiran B.4.
7. Nilai hasil berupa persentase akurasi dari masing-masing dataset yang digunakan.
8. Nilai maksimum yang didapatkan (Maks).

Tabel 6.4 menampilkan bahwa pada iterasi 2 dengan nilai *learning rate* = 0.1 dan *decrement* = 0.3 sudah terjadi konvergensi, bahkan pada *dataset* 1, 3, 4 terjadi konvergen dari iterasi 1.

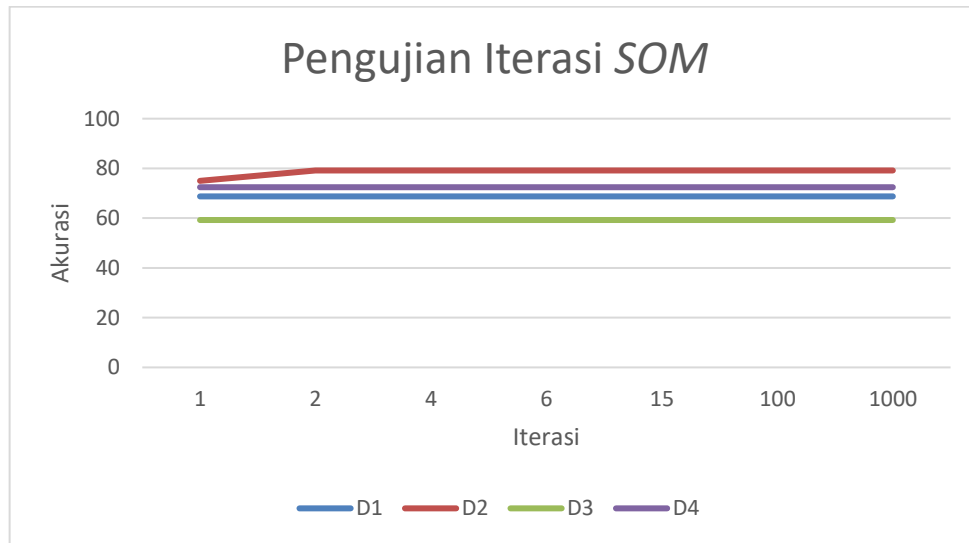
Tabel 6.4 Pengujian Iterasi Pada *SOM*

Iterasi	D1	D2	D3	D4	Maks
1	68.75	75	59.259	72.436	75
2	68.75	79.167	59.259	72.436	79.167
4	68.75	79.167	59.259	72.436	79.167
6	68.75	79.167	59.259	72.436	79.167
15	68.75	79.167	59.259	72.436	79.167

Tabel 6.4 Pengujian Iterasi Pada *SOM* (Lanjutan)

Iterasi	D1	D2	D3	D4	Maks
100	68.75	79.167	59.259	72.436	79.167
1000	68.75	79.167	59.259	72.436	79.167

Gambar 6.3 merupakan grafik pengujian iterasi pada pelatihan *SOM*. Grafik tersebut menunjukkan bahwa pada *dataset* 1, 3 dan 4 iterasi 1 hingga 1000 tidak mengalami perubahan nilai atau sudah konvergen, namun pada *dataset* 2 titik konvergen didapatkan pada iterasi 2.



Gambar 6.3 Grafik Pengujian Iterasi *SOM*

6.3 Pengujian LVQ

Pengujian pada *LVQ* dilakukan untuk menentukan semua komponen terbaik sesuai dengan apa yang sudah dirancang agar dapat dibandingkan dengan kombinasi algoritme *SOM-LVQ*. Pengujian yang dilakukan antara lain pengujian *Learning Rate*, *Decrement Learning Rate* dan Iterasi.

6.3.1 Pengujian *Learning Rate* Pada *LVQ*

Tabel 6.5 merupakan uji coba untuk menentukan *learning rate* terbaik pada pelatihan *LVQ*. *Decrement learning rate* awal bernilai 0.5 (Azmi, 2014). Iterasi yang digunakan adalah 100 iterasi. Unsur-unsur yang digunakan untuk pengujian ini antara lain:

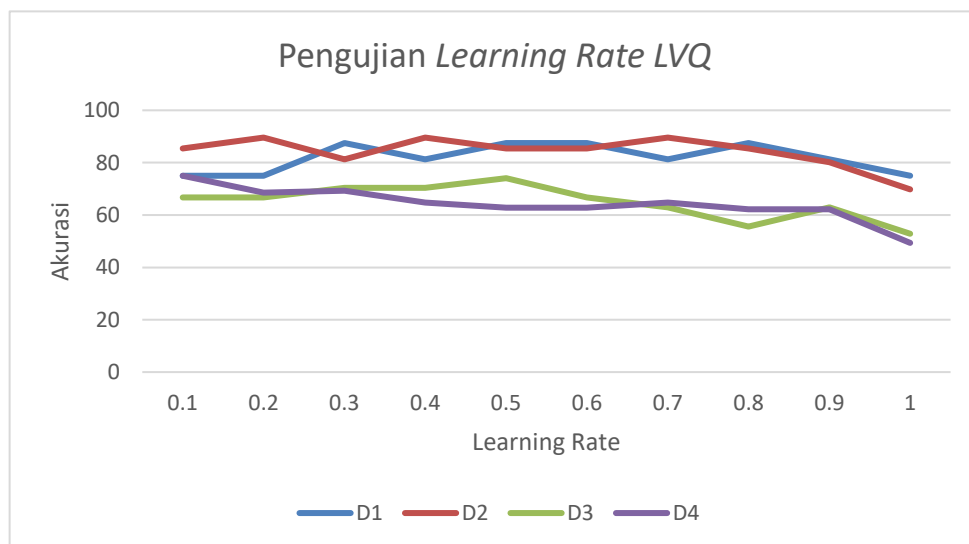
1. Rentang nilai *alpha/learning rate* dari 0.1 hingga 1.
2. *Dataset* awal yang menggunakan jumlah kasus (D1). Data terdapat pada Lampiran B.1.
3. *Dataset* awal utuh tanpa adanya jumlah kasus (D2). Data terdapat pada Lampiran B.2.
4. *Dataset* lanjut yang menggunakan jumlah kasus (D3). Data terdapat pada Lampiran B.3.

5. *Dataset* lanjut utuh tanpa adanya jumlah kasus (D4). Data terdapat pada Lampiran B.4.
6. Nilai hasil berupa persentase akurasi dari masing-masing dataset yang digunakan.
7. Nilai maksimum yang didapatkan (Maks).

Tabel 6.5 menampilkan hasil pengujian *learning rate* pada pelatihan LVQ dari keseluruhan data yang digunakan. Nilai *learning rate* 0,2, 0,4 dan 0,7 pada Dataset 2 (D2) mendapatkan nilai akurasi yang sama yaitu 89.583% dan nilai tersebut juga merupakan nilai maksimum dari keseluruhan nilai akurasi yang didapat. Nilai *learning rate* 0,2, 0,4 dan 0,7 adalah variabel terbaik.

Tabel 6.5 Pengujian *Learning Rate* LVQ

Learning rate	D1	D2	D3	D4	Maks
0.1	75	85.416	66.67	75	85.416
0.2	75	89.583	66.67	68.589	89.583
0.3	87.5	81.25	70.37	69.231	87.5
0.4	81.25	89.583	70.37	64.746	89.583
0.5	87.5	85.417	74.074	62.82	87.5
0.6	87.5	85.417	66.67	62.82	87.5
0.7	81.25	89.583	62.963	64.746	89.583
0.8	87.5	85.417	55.56	62.179	87.5
0.9	81.25	80.208	62.963	62.179	81.25
1	75	69.792	52.852	49.359	75
Nilai Maks					89.583



Gambar 6.4 Grafik Pengujian *Learning Rate* LVQ

Gambar 6.4 merupakan grafik pengujian *learning rate* pada pelatihan LVQ. Grafik tersebut menunjukkan bahwa nilai *learning rate* 0,2, 0,4 dan 0,7 pada dataset 2 (D2) menghasilkan nilai yang sama dan juga mencapai nilai maksimum.

Dataset 1 juga memberikan hasil yang mendekati nilai maksimum dengan nilai *learning rate* 0,3 hingga 0,9.

6.3.2 Pengujian *Decrement Learning Rate* Pada *LVQ*

Tabel 6.6 merupakan pengujian untuk menentukan *decrement learning rate* terbaik pada pelatihan *LVQ*. *Learning rate* yang digunakan adalah 0,2 dikarenakan pada pengujian *learning rate* pelatihan *LVQ* nilai 0,2 yang memperoleh nilai maksimum. Iterasi yang digunakan adalah 100 iterasi. Unsur-unsur yang digunakan untuk pengujian ini antara lain:

1. Rentang nilai *decrement learning rate* dari 0,1 hingga 1.
2. *Dataset* awal yang menggunakan jumlah kasus (D1). Data terdapat pada Lampiran B.1.
3. *Dataset* awal utuh tanpa adanya jumlah kasus (D2). Data terdapat pada Lampiran B.2.
4. *Dataset* lanjut yang menggunakan jumlah kasus (D3). Data terdapat pada Lampiran B.3.
5. *Dataset* lanjut utuh tanpa adanya jumlah kasus (D4). Data terdapat pada Lampiran B.4.
6. Nilai hasil berupa persentase akurasi dari masing-masing dataset yang digunakan.
7. Nilai maksimum yang didapatkan (Maks).

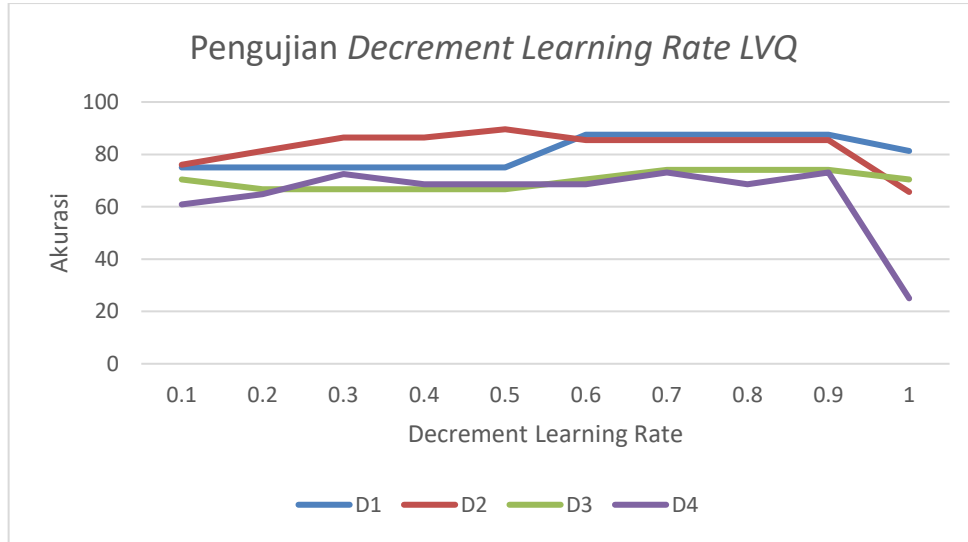
Tabel 6.6 menampilkan hasil pengujian *decrement learning rate* pada pelatihan *LVQ* dari keseluruhan data yang digunakan. Nilai *decrement learning rate* 0,5 pada *dataset 2* (D2) mendapatkan nilai maksimum dari keseluruhan nilai akurasi yang didapat yaitu 89.583%. Nilai *decrement learning rate* 0.5 adalah variabel terbaik.

Tabel 6.6 Pengujian *Decrement LVQ*

Decrement	D1	D2	D3	D4	Maks
0.1	75	76.042	70.37	60.897	76.042
0.2	75	81.25	66.67	64.744	81.25
0.3	75	86.458	66.67	72.44	86.458
0.4	75	86.458	66.67	68.589	86.458
0.5	75	89.583	66.67	68.589	89.583
0.6	87.5	85.416	70.37	68.589	87.5
0.7	87.5	85.416	74.074	73.077	87.5
0.8	87.5	85.416	74.074	68.589	87.5
0.9	87.5	85.416	74.074	73.077	87.5
1	81.25	65.625	70.37	25	81.25
				Nilai Maks	89.583

Gambar 6.5 merupakan grafik pengujian *decrement learning rate* pada pelatihan *LVQ*. Grafik tersebut menunjukkan bahwa nilai *decrement* 0,5 pada *dataset 2* (D2) meraih nilai maksimum yaitu 89,583%. *Dataset 1* (D1) juga

memberikan hasil yang mendekati nilai maksimum yaitu 87,5% dengan nilai *decrement* 0,6 hingga 0,9.



Gambar 6.5 Grafik Pengujian Decrement LVQ

6.3.3 Pengujian Iterasi Pada LVQ

Tabel 6.7 merupakan pengujian untuk menentukan iterasi terbaik pada pelatihan LVQ. Unsur-unsur yang digunakan untuk pengujian ini antara lain:

1. Rentang iterasi yang diuji.
2. Nilai *learning rate* = 0,2 dan *decrement* = 0,5 sesuai dengan hasil terbaik pengujian sebelumnya.
3. *Dataset* awal yang menggunakan jumlah kasus (D1). Data terdapat pada Lampiran B.1.
4. *Dataset* awal utuh tanpa adanya jumlah kasus (D2). Data terdapat pada Lampiran B.2.
5. *Dataset* lanjut yang menggunakan jumlah kasus (D3). Data terdapat pada Lampiran B.3.
6. *Dataset* lanjut utuh tanpa adanya jumlah kasus (D4). Data terdapat pada Lampiran B.4.
7. Nilai hasil berupa persentase akurasi dari masing-masing dataset yang digunakan.
8. Nilai maksimum yang didapatkan (Maks).

Tabel 6.7 menampilkan bahwa pada iterasi 6 dengan nilai *learning rate* = 0.2 dan *decrement* = 0.5 sudah terjadi konvergensi pada keseluruhan data, bahkan pada *dataset* 1 tidak terjadi perubahan sejak iterasi 1, *dataset* 3 sejak iterasi 2 dan *dataset* 4 sejak iterasi 4.

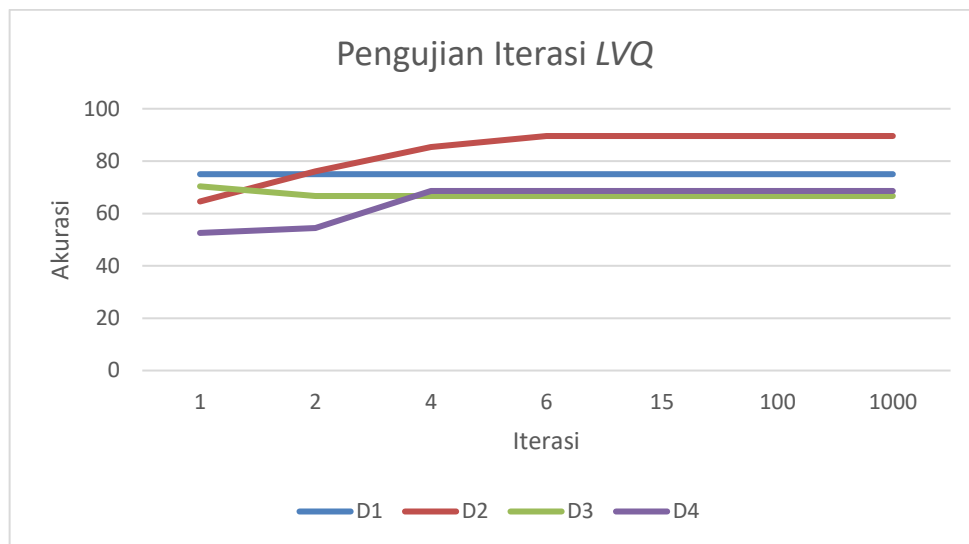
Tabel 6.7 Pengujian Iterasi Pada LVQ

Iterasi	D1	D2	D3	D4	Maks
1	75	64.583	70.37	52.564	75
2	75	76.042	66.67	54.487	76.042

Tabel 6.7 Pengujian Iterasi Pada LVQ

Iterasi	D1	D2	D3	D4	Maks
4	75	85.417	66.67	68.589	85.417
6	75	89.583	66.67	68.589	89.583
15	75	89.583	66.67	68.589	89.583
100	75	89.583	66.67	68.589	89.583
1000	75	89.583	66.67	68.589	89.583

Gambar 6.6 merupakan grafik pengujian iterasi pada pelatihan LVQ. Grafik tersebut menunjukkan bahwa pada iterasi 6 semua *dataset* sudah tidak mengalami perubahan. *Dataset 2* (D2) mencapai nilai maksimum dan sudah konvergen pada iterasi 6 dengan nilai 89,583%. *Dataset* lain sudah mencapai konvergen pada iterasi sebelumnya.



Gambar 6.6 Grafik Pengujian Iterasi LVQ

6.4 Pengujian SOM-LVQ

Pengujian pada SOM-LVQ dilakukan untuk menentukan semua komponen terbaik sesuai dengan apa yang sudah dirancang agar dapat dibandingkan dengan kombinasi algoritme LVQ. Pada pengujian SOM-LVQ perlu dipertimbangkan juga komponen terbaik dari pengujian SOM sebagai penentu bobot awal dan LVQ sebagai proses pelatihan akhir. Pengujian yang dilakukan antara lain pengujian *Learning Rate*, *Decrement Learning Rate* dan Iterasi.

6.4.1 Pengujian *Learning Rate* Pada SOM-LVQ

Tabel 6.8 merupakan uji coba untuk menentukan *learning rate* terbaik pada pelatihan SOM-LVQ. Bobot yang digunakan adalah hasil pelatihan dari SOM dengan iterasi dan variabel terbaik sesuai dengan pengujian pada SOM yaitu *learning rate*= 0,1 dan *decrement*= 0,3 , lalu dilakukan pelatihan LVQ dengan iterasi dan variabel terbaik sesuai dengan pengujian pada LVQ yaitu *decrement* = 0,5. Unsur-unsur yang digunakan untuk pengujian ini antara lain:

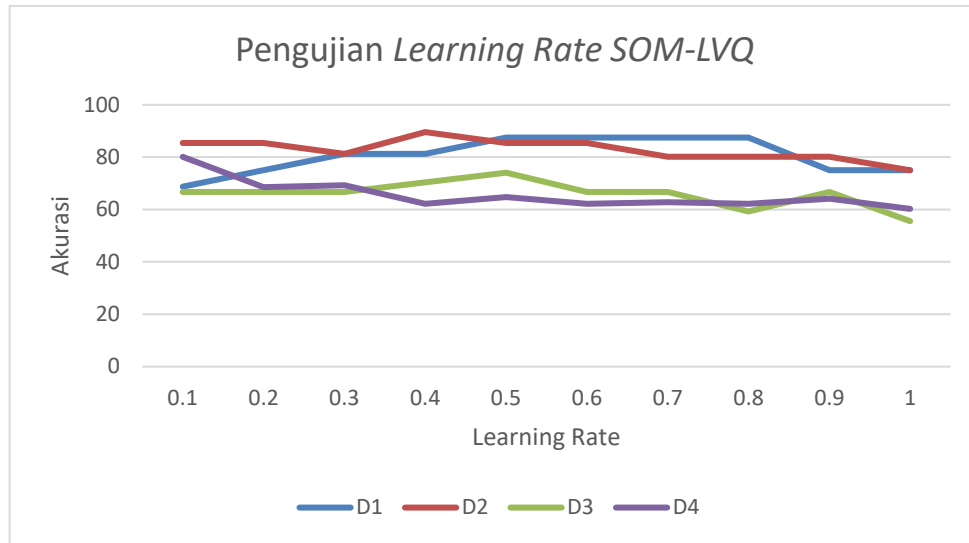
1. Rentang nilai *alpha/learning rate* dari 0,1 hingga 1.
2. *Dataset* awal yang menggunakan jumlah kasus (D1). Data terdapat pada Lampiran B.1.
3. *Dataset* awal utuh tanpa adanya jumlah kasus (D2). Data terdapat pada Lampiran B.2.
4. *Dataset* lanjut yang menggunakan jumlah kasus (D3). Data terdapat pada Lampiran B.3.
5. *Dataset* lanjut utuh tanpa adanya jumlah kasus (D4). Data terdapat pada Lampiran B.4.
6. Nilai hasil berupa persentase akurasi dari masing-masing dataset yang digunakan.
7. Nilai maksimum yang didapatkan (Maks).

Tabel 6.8 menampilkan hasil pengujian *learning rate* pada pelatihan SOM-LVQ dari keseluruhan data yang digunakan. Nilai *learning rate* 0,4 pada *dataset* 2 (D2) mendapatkan nilai akurasi 89.583% dan nilai tersebut juga merupakan nilai maksimum dari keseluruhan nilai akurasi yang didapat. Nilai *learning rate* 0,4 adalah variabel terbaik.

Tabel 6.8 Pengujian *Learning Rate* SOM-LVQ

Learning rate	D1	D2	D3	D4	Maks
0.1	68.75	85.417	66.67	80.128	85.417
0.2	75	85.417	66.67	68.589	85.417
0.3	81.25	81.25	66.67	69.231	81.25
0.4	81.25	89.583	70.37	62.179	89.583
0.5	87.5	85.417	74.074	64.744	87.5
0.6	87.5	85.417	66.67	62.179	87.5
0.7	87.5	80.208	66.67	62.82	87.5
0.8	87.5	80.208	59.259	62.179	87.5
0.9	75	80.208	66.67	64.103	80.208
1	75	75	55.56	60.256	75
Nilai Maks					89.583

Gambar 6.7 merupakan grafik pengujian *learning rate* pada pelatihan SOM-LVQ. Grafik tersebut menunjukkan bahwa nilai *learning rate* *learning rate* 0,4 pada *dataset* 2 (D2) mencapai nilai maksimum yaitu 89.583%. *Dataset* 1 (D1) juga mendekati nilai maksimum dengan *learning rate* 0,5 hingga 0,8 yaitu 87,5%.



Gambar 6.7 Grafik Pengujian *Learning Rate* SOM-LVQ

6.4.2 Pengujian *Decrement Learning Rate* Pada SOM-LVQ

Tabel 6.9 merupakan uji coba untuk menentukan *decrement learning rate* terbaik pada pelatihan SOM-LVQ. Bobot yang digunakan adalah hasil pelatihan dari SOM dengan iterasi dan variabel terbaik sesuai dengan pengujian pada SOM yaitu *learning rate*= 0,1 dan *decrement*= 0,3, lalu dilakukan pelatihan LVQ dengan iterasi dan variabel terbaik sesuai dengan pengujian *learning rate* terbaik pada SOM-LVQ yaitu *learning rate*= 0,4. Unsur-unsur yang digunakan untuk pengujian ini antara lain:

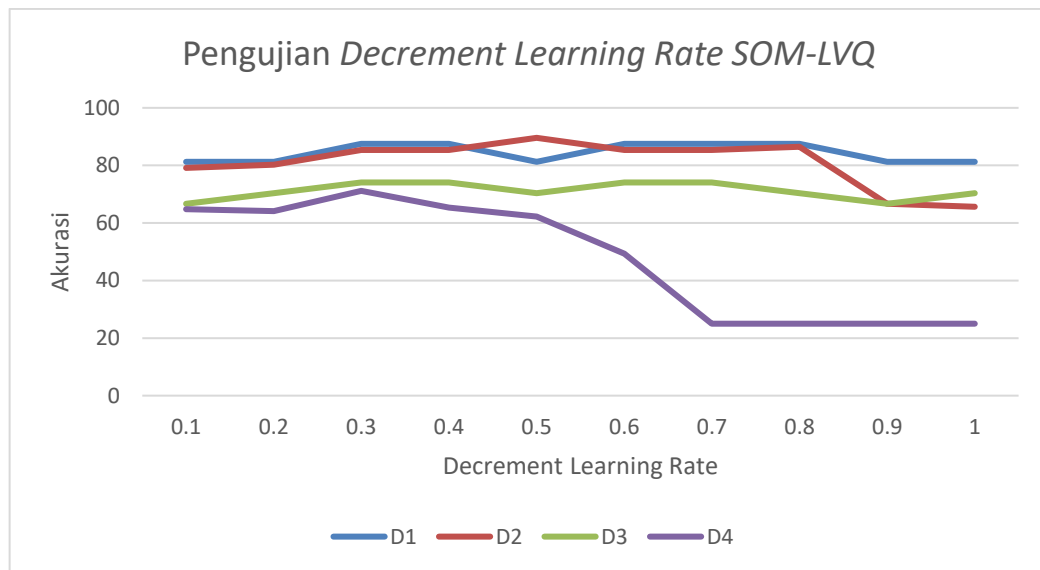
1. Rentang nilai *alpha/learning rate* dari 0,1 hingga 1.
2. *Dataset* awal yang menggunakan jumlah kasus (D1). Data terdapat pada Lampiran B.1.
3. *Dataset* awal utuh tanpa adanya jumlah kasus (D2). Data terdapat pada Lampiran B.2.
4. *Dataset* lanjut yang menggunakan jumlah kasus (D3). Data terdapat pada Lampiran B.3.
5. *Dataset* lanjut utuh tanpa adanya jumlah kasus (D4). Data terdapat pada Lampiran B.4.
6. Nilai hasil berupa persentase akurasi dari masing-masing dataset yang digunakan.
7. Nilai maksimum yang didapatkan (Maks).

Tabel 6.9 menampilkan hasil pengujian *decrement learning rate* pada pelatihan SOM-LVQ dari keseluruhan data yang digunakan. Nilai *decrement learning rate* 0,5 pada *dataset* 2 (D2) mendapatkan nilai maksimum dari keseluruhan nilai akurasi yang didapat yaitu 89.583%. Nilai *decrement learning rate* 0.5 adalah variabel terbaik.

Tabel 6.9 Pengujian *Decrement SOM-LVQ*

Decrement	D1	D2	D3	D4	Maks
0.1	81.25	79.167	66.67	64.744	81.25
0.2	81.25	80.208	70.37	64.103	81.25
0.3	87.5	85.417	74.074	71.154	87.5
0.4	87.5	85.417	74.074	65.385	87.5
0.5	81.25	89.583	70.37	62.179	89.583
0.6	87.5	85.417	74.074	49.359	87.5
0.7	87.5	85.417	74.074	25	87.5
0.8	87.5	86.458	70.37	25	87.5
0.9	81.25	66.67	66.67	25	81.25
1	75	25	25.926	25	75
Nilai Maks					89.583

Gambar 6.8 merupakan grafik pengujian *decrement learning rate* pada pelatihan *SOM-LVQ*. Grafik tersebut menunjukkan bahwa nilai *decrement* 0,5 pada *dataset 2* (D2) meraih nilai maksimum yaitu 89,583%. *Dataset 1* (D1) juga memberikan hasil yang mendekati nilai maksimum yaitu 87,5% dengan nilai *decrement* 0,3 hingga 0,8.

Gambar 6.8 Grafik Pengujian *Decrement SOM-LVQ*

6.4.3 Pengujian Iterasi Pada *SOM-LVQ*

Tabel 6.10 merupakan pengujian untuk menentukan iterasi terbaik pada pelatihan *SOM-LVQ*. Unsur-unsur yang digunakan untuk pengujian ini antara lain:

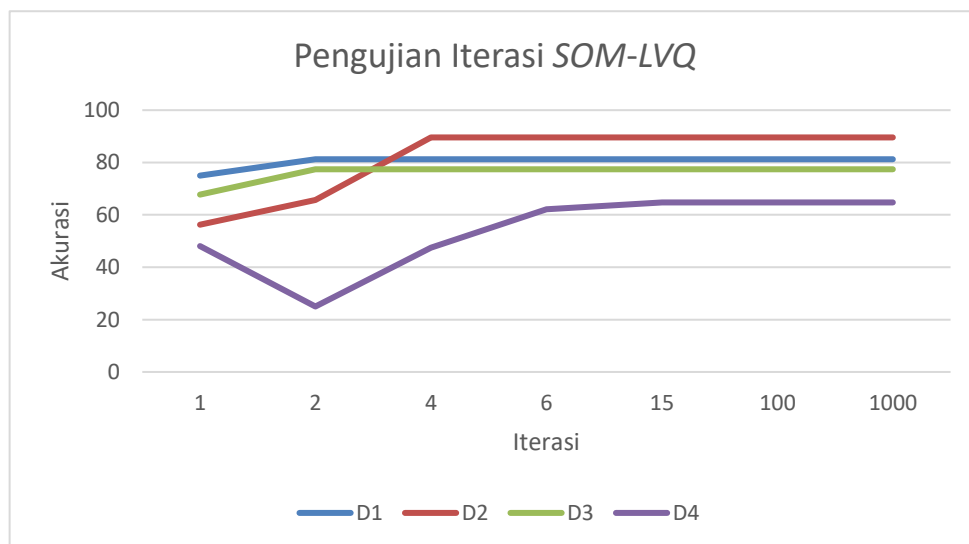
1. Rentang iterasi yang diuji.
2. Nilai *learning rate* = 0,4 dan *decrement* = 0,5 sesuai dengan hasil terbaik pengujian sebelumnya.

3. *Dataset* awal yang menggunakan jumlah kasus (D1). Data terdapat pada Lampiran B.1.
4. *Dataset* awal utuh tanpa adanya jumlah kasus (D2). Data terdapat pada Lampiran B.2.
5. *Dataset* lanjut yang menggunakan jumlah kasus (D3). Data terdapat pada Lampiran B.3.
6. *Dataset* lanjut utuh tanpa adanya jumlah kasus (D4). Data terdapat pada Lampiran B.4.
7. Nilai hasil berupa persentase akurasi dari masing-masing dataset yang digunakan.
8. Nilai maksimum yang didapatkan (Maks).

Tabel 6.10 menampilkan bahwa pada iterasi 15 dengan nilai *learning rate* = 0.4 dan *decrement* = 0.5 sudah terjadi konvergensi pada keseluruhan data, namun nilai maksimum sudah didapat pada *dataset* 2 (D2) dengan iterasi 4.

Tabel 6.10 Pengujian Iterasi SOM-LVQ

Iterasi	D1	D2	D3	D4	Maks
1	75	56.25	67.742	48.077	75
2	81.25	65.625	77.419	25	81.25
4	81.25	89.583	77.419	47.44	89.583
6	81.25	89.583	77.419	62.179	89.583
15	81.25	89.583	77.419	64.744	89.583
100	81.25	89.583	77.419	64.744	89.583
1000	81.25	89.583	77.419	64.744	89.583



Gambar 6.9 Grafik Pengujian Iterasi SOM-LVQ

Gambar 6.9 merupakan grafik pengujian iterasi pada pelatihan SOM-LVQ. Grafik tersebut menunjukkan bahwa pada iterasi 15 semua *dataset* sudah tidak mengalami perubahan. *Dataset* 2 (D2) mencapai nilai maksimum dan sudah konvergen pada iterasi 4 dengan nilai 89,583%. *Dataset* lain sudah mencapai

konvergen pada iterasi sebelumnya. *Dataset* 1 dan 3 bahkan mengalami konvergen pada iterasi 2. *Dataset* 4 konvergen pada iterasi 15 dan merupakan titik konvergen terakhir dari seluruh data.

6.5 Analisis dan Perbandingan

Semua pengujian sudah dilakukan terhadap algoritme yang digunakan, maka dilakukan analisis terhadap hasil pengujian. Hal yang paling signifikan untuk dianalisis adalah konvergensi dini pada iterasi kecil. Hal tersebut dikarenakan data yang digunakan pada penelitian ini adalah data diskrit atau biner. Data diskrit hanya memiliki nilai 0 dan 1 yang menyebabkan nilai perbaikan bobot pada pelatihan *SOM*, *LVQ* dan *SOM-LVQ* menjadi cepat konvergen. Unsur-unsur yang ada pada perbaikan bobot seperti *learning rate* dan *decrement* juga mempengaruhi perubahan bobot hasil perbaikan dikarenakan pada iterasi tertentu jika nilai *learning rate* sudah sangat kecil maka semakin tidak mempengaruhi dalam perubahan/perbaikan bobot. Hal tersebut menyebabkan pengelompokan/klasifikasi data menjadi konvergen dikarenakan bobot yang berfungsi sebagai perwakilan kelas tidak berubah secara signifikan. Hal lain yang mempengaruhi hasil dan juga tingkat akurasi adalah pemilihan kombinasi *learning rate* dan *decrement learning rate* yang sesuai. *Learning rate* pada algoritme *SOM*, *LVQ* dan *SOM-LVQ* memiliki fungsi sebagai nilai perbaikan bobot, lalu nilai pengurang/*decrement learning rate* berfungsi untuk memodifikasi/*update* nilai *learning rate* pada setiap iterasi dan berpengaruh pada pelatihan dikarenakan nilai *learning rate* mempengaruhi perubahan bobot. Kecocokan nilai pengurang terhadap laju pembelajaran pada pelatihan mempengaruhi pelatihan dan juga hasil akhir.

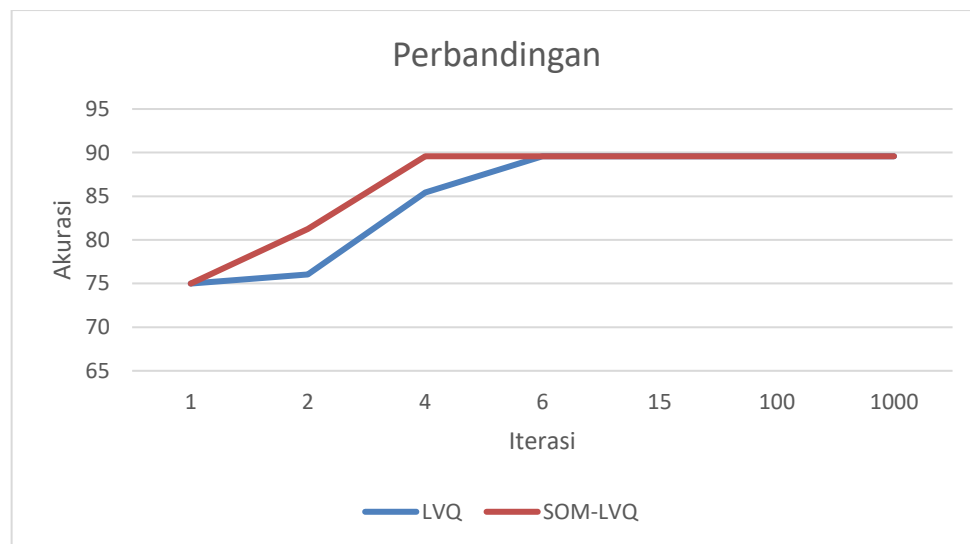
Perbandingan antara algoritme *LVQ* dan *SOM-LVQ* dilakukan agar mengetahui perbedaan akurasi maksimum yang dihasilkan dan juga membuktikan apakah kombinasi algoritme *SOM-LVQ* lebih baik dari *LVQ* murni pada penelitian ini. Tabel 6.11 merupakan Perbandingan hasil akhir dari nilai maksimum yang didapatkan pada pengujian masing-masing algoritme beserta dataset yang mencapai nilai maksimum pada iterasi yang ditentukan. Tabel 6.11 memaparkan bahwa *dataset* 2 (D2) merupakan *dataset* yang mencapai nilai maksimum yang didapatkan dari kedua algoritme yaitu 89,583% dan juga *dataset* yang lebih mendominasi dalam hal akurasi dibanding *dataset* lainnya. Tabel 6.11 juga menampilkan bahwa *SOM-LVQ* lebih cepat mendapatkan nilai maksimum daripada *LVQ* murni, yaitu pada iterasi 4.

Tabel 6.11 Perbandingan Hasil Akhir

Iterasi	Maks LVQ	Dataset	Maks SOM-LVQ	Dataset
1	75	D1	75	D1
2	76.042	D2	81.25	D1
4	85.417	D2	89.583	D2
6	89.583	D2	89.583	D2
15	89.583	D2	89.583	D2

Iterasi	Maks LVQ	Dataset	Maks SOM-LVQ	Dataset
100	89.583	D2	89.583	D2
1000	89.583	D2	89.583	D2

Gambar 6.10 merupakan grafik dari nilai maksimum yang didapat pada pengujian *LVQ* pada Tabel 6.7 dan pengujian *SOM-LVQ* pada Tabel 6.10. Gambar tersebut menjelaskan bahwa nilai akurasi dari *SOM-LVQ* dari awal lebih bagus dan lebih cepat meraih nilai maksimum daripada *LVQ* murni. Pencapaian nilai akurasi maksimum dari kedua algoritme yaitu sama-sama 89,583%, namun algoritme *SOM-LVQ* lebih cepat mendapat nilai maksimum yaitu pada iterasi 4. Hal tersebut dikarenakan bobot yang digunakan pada *SOM-LVQ* telah dilatih menggunakan algoritme *SOM*, sedangkan bobot pada *LVQ* murni menggunakan data latih yang belum dilatih.



Gambar 6.10 Grafik Perbandingan

BAB 7 PENUTUP

7.1 Kesimpulan

Kesimpulan yang didapatkan dari hasil pengujian dalam penelitian ini adalah sebagai berikut:

1. Kombinasi algoritme *Self Organizing Maps (SOM)* dan *Learning Vector Quantization (LVQ)* dapat diimplementasikan untuk klasifikasi jenis audio berdasarkan kondisi psikologi dan jenis kegiatan. Kondisi psikologi dan jenis kegiatan dijadikan fitur, lalu jenis audio dijadikan kelas tujuan/label data. Data diimplementasikan menjadi 11 fitur dan 4 kelas pada penelitian ini. Fitur diimplementasikan dalam bentuk biner. Angka 1 merepresentasikan ya, angka 0 merepresentasikan tidak. Satu vektor data berisi 11 fitur dan 1 nilai kelas/label kelas yang kemudian dilakukan proses pelatihan.
2. Nilai akurasi maksimal yang didapatkan pada penelitian ini adalah 89,583%. Kombinasi algoritme *SOM-LVQ* mencapai nilai akurasi maksimal tersebut dengan 4 iterasi pelatihan, sedangkan *LVQ* murni 6 iterasi maksimal. Walaupun dengan nilai akurasi yang sama, *SOM-LVQ* lebih baik dalam kecepatan mendapatkan nilai optimal (iterasi lebih kecil). *Dataset 2 (D2)* merupakan *dataset* yang mencapai nilai maksimum dari kedua algoritme yaitu 89,583% dan juga *dataset* yang lebih mendominasi dalam hal akurasi dibanding *dataset* lainnya. Hal tersebut dikarenakan pada *dataset 2* terdapat data kembar yang menyebabkan nilai akurasi pada saat pengujian menjadi baik. *Dataset 2* sesungguhnya sama dengan *dataset 1*, namun pada *dataset 1* data yang sama dijadikan jumlah kasus. Tujuan digunakan beberapa variasi *dataset* adalah untuk mengetahui bentuk dataset seperti apa yang cocok diterapkan pada penelitian ini.

7.2 Saran

Saran yang diberikan dari penelitian ini untuk penelitian selanjutnya adalah sebagai berikut:

1. Perlu dilakukan penelitian yang lebih lanjut mengenai teknik pemilihan bobot awal pada algoritme *SOM* guna memberikan hasil bobot yang lebih baik pada pelatihan *SOM-LVQ*.
2. Perlu dilakukan penelitian lebih lanjut mengenai teknik pengumpulan data dan pemilihan fitur yang lebih valid dalam melakukan klasifikasi pada kasus penelitian ini. Nilai data yang digunakan bisa berupa data kuantitatif dengan derajat interval tertentu bukan hanya nilai biner (ya atau tidak) saja. Validasi untuk pemilihan fitur dan pengumpulan data juga perlu didukung dari publikasi yang berkaitan, penelitian sebelumnya ataupun informasi lebih banyak dari beberapa pakar yang berkaitan. Variasi data dan juga jumlah data perlu ditambah.

DAFTAR PUSTAKA

- Azmi, M., 2014. Komparasi Metode Jaringan Syaraf Tiruan SOM dan LVQ Untuk Mengidentifikasi Data Bunga Iris. *Jurnal TEKNOIF*, Volume 2, p. 65.
- Budianita, E. & Arni, U. D., 2015. Penerapan Learning Vector Quantization Penentuan Bidang Konsentrasi Tugas Akhir (Studi Kasus: Mahasiswa Teknik Informatika UIN Suska Riau). Volume 1, pp. 1-5.
- Chen, B., Xiao, C. & Song, X., 2012. *Intelligent Decision System Based on SOM-LVQ Neural Network for Soccer Robot*. Xi'an, IEEE.
- E., 2013. *Pengaruh Mendengarkan Murottal Q.S. Ar-Rahman Terhadap Pola Tekanan Darah Pada Pasien Hipertensi di Rumah Sakit Nur Hidayah Yogyakarta*, Yogyakarta: Universitas Muhammadiyah Yogyakarta.
- Handayani, R., Fajarsari, D., Asih, D. R. T. & Rohmah, D. N., 2014. Pengaruh Terapi Murottal Al-Qur'an Untuk Penurunan Nyeri Persalinan dan Kecemasan Pada Ibu Bersalin Kala I Fase Aktif. *Jurnal Ilmiah Kebidanan*, Volume 5, pp. 1-15.
- Hilmi, M. S. N., 2015. *Makalah Multimedia Suara dan Audio*, Purwokerto: s.n.
- Ifadah, M. & Aimah, S., 2012. Keefektifan Lagu Sebagai Media Belajar Dalam Pengajaran Pronunciation/Pengucapan.
- Mills, B. D., 1996. Effects of Music on Assertive Behavior During Exercise by Middle-School Age.
- Nagara, R. M. S., 2016. Keterkaitan Logika dengan Emosi dan Perasaan. pp. 1-5.
- Nugroho, A. P., 2013. Musik dan Psikologi. *Jurnal Musik dan Psikologi*, pp. 1-6.
- Pakaja, F., Naba, A. & P., 2012. Peramalan Penjualan Mobil Menggunakan Jaringan Syaraf Tiruan dan Certainty Factor. pp. 1-6.
- Putri, L. A. A. R. & Hartati, S., 2016. Klasifikasi Genre Musik Menggunakan Learning Vector Quantization dan Self Organizing Map. Volume 9, pp. 1-9.
- Sailer, U. & Hassenzahl, M., 2000. Assessing Noise Annoyance: An Improvement-Oriented. Volume 43.
- Sakrie, D., 2015. *100 Tahun Musik Indonesia*. Jakarta: Gagas Media.
- Sari, A. P. & Grashinta, A., 2015. Pengaruh Jenis Musik Terhadap Performa Kognitif yang Menuntut Ingatan Jangka Pendek Pada Anak-Anak Usia 7-11 Tahun. Volume 2, pp. 1-11.
- Supradewi, R., 2010. Otak, Musik dan Proses Belajar. Volume 18, pp. 58-68.
- Widhyatama, S., 2012. *Sejarah Musik dan Apresiasi Seni*. Jakarta: PT Balai Pustaka.
- Yuriastien, E., 2017. *Pengertian Psikologi* [Interview] (1 November 2017).

LAMPIRAN A HASIL WAWANCARA

A.1 Wawancara Pakar

Pada penelitian ini pakar adalah seorang psikolog dari RSJ. dr. Radjiman Wediodiningrat Lawang yang bernama Effiana Yuriastien, S.Si, Psikolog. Berikut beberapa tanya jawab yang telah tercatat:

1. Pertanyaan:

Apakah pengertian dari psikolog dan psikologi?

Jawaban:

Psikolog adalah profesi yang membidangi ilmu kejiwaan, sedangkan psikologi adalah ilmu tentang kejiwaan.

2. Pertanyaan:

Apakah hubungan psikologi dengan perasaan atau emosi manusia?

Jawaban:

Semua itu adalah aspek dari psikologi, semua macam perasaan itu adalah emosi.

3. Pertanyaan:

Apakah jenis dari emosi manusia dapat disebutkan atau dijelaskan nama-namanya?

Jawaban:

Tidak dapat dijabarkan secara pasti, namun memang ada banyak seperti senang dan tidak senang.

4. Pertanyaan:

Apakah musik dan murottal Qur'an dapat dijadikan cara untuk memperbaiki kondisi psikologi seseorang?

Jawaban:

Itu merupakan salah satu pendekatannya, bisa juga dijadikan sarana untuk terapi relaksasi.

5. Pertanyaan:

Apakah menurut anda pemilihan kondisi psikologi untuk dijadikan fitur pada penelitian saya ini sudah cukup baik? (memberikan kuisioner).

Jawaban:

Baik atau tidaknya tidak dapat saya pastikan, hanya saja emosi memang memiliki banyak macam seperti ini.

6. Pertanyaan:

Untuk pengumpulan data apakah lebih baik menggunakan biner (1/0)(ya/tidak) atau *range* nilai seperti 1 s/d 5.

Jawaban:

Tidak dapat divalidasi jika menggunakan *range* nilai, lebih baik biner.

A.1 Wawancara Pengumpulan Data

Pengumpulan data pada penelitian ini dilakukan secara langsung (wawancara langsung) ataupun melalui perantara *social media*. Tabel dibawah merupakan kuisisioner yang digunakan untuk mengumpulkan data.

-Isilah kolom yang sudah disediakan sesuai dengan apa yang anda alami dan rasakan jika anda mendengarkan jenis audio tersebut (boleh menggunakan v atau x).

jenis audio yang didengarkan	Kondisi Psikologi (emosi)								Jenis Kegiatan		
	konsentrasi	tenang	semangat	ceria	bebas	bertenaga	sendu	sestengah	Kegiatan Berfokus	Berolahraga	Istirahat
klasik											
Murottal											
rock											
pop											

Proses wawancara secara langsung maupun melalui *social media* kurang lebih seperti ini (dalam lingkup teman, bahasa yang digunakan semi formal):

1. Pertanyaan:

Selamat pagi Ninda, maaf apakah aku boleh bertanya sebentar untuk keperluan data skripsi?

Jawaban:

Pagi Han, wah iya boleh, mau nanya apa?

2. Pertanyaan:

Aku punya 4 genre musik yg aku jadiin penelitian, yaitu Klasik, Murottal, Rock dan Pop. dari masing masing genre tersebut kan punya karakteristik yang berbeda-beda menurutku dan si pendengar juga pasti merasakan perasaan yang berbeda beda jika mendengarkan salah satu diantaranya, nah sementara ini menurutku, aku sih ngerasain perasaan seperti ini jika aku mendengarkan musik-musik ini:

Klasik: Konsentrasi, fokus

Murottal: Tenang, istirahat

Rock: Bebas, semangat, bertenaga buat olahraga

Pop: Sendu/galau, untuk istirahat atau santai (jika yg lagu galau)

nah kalau menurutmu sendiri, dari masing2 genre tersebut, perasaan apa yang kamu rasain? Mungkin aku bisa bantu dengan tabel kuisisioner ini untuk memberi pilihan emosi apa saja (memberikan kuisisioner berbentuk *print out* atau gambar seperti pada tabel diatas).

Jawaban:

Oke Han kucoba jawab sesuai pendapatku ya

Klasik: Biasanya aku dengerin pas mau tidur sih **tenang**, atau momen momen tertentu pas lagi pusing buat **istirahat** (biasanya dengerin instrumen

piano / gitar). Mungkin juga kondisi ini sering dengerin musik instrumen (saxophone instrumen / piano) biasa bikin **konsentrasi** sama **tenang**.

Murottal: nyaman kalau didengerin lagi **istirahat**, bikin **tenang**.

Rock: **Bebas**, biasanya aku juga dengerin rock pas situasi **perasaan lagi naik bertenaga, semangat** buat **olahraga**.

Pop: pop sendiri juga bisa didengerin pas lagi **santai**, tergantung jenis lagunya kalau lagu yang sedih ya **sendu/galau**.

kalau itu sih biasanya yg aku rasain.

Maka data yang didapatkan dari wawancara diatas seperti ini:

Jenis audio yang didengarkan	Kondisi Psikologi (emosi)								Jenis Kegiatan		
	konsentrasi	tenang	semangat	ceria	bebas	bertenaga	sendu	santai	Kegiatan Berfokus	Berolahraga	Istirahat
klasik	1	1									1
Murottal		1									1
rock			1		1	1				1	
pop							1				1

LAMPIRAN B DATA

B.1 Data Awal (D1)

Data awal atau *dataset* 1 (D1) adalah data awal yang terkumpul dengan beberapa jumlah kasus dari tiap data yang memiliki kesamaan.

D A T A	Juml ah Kasus	FITUR											KEL AS
		kons entr asi	te na ng	sem ang at	c er ia	be ba s	bert ena ga	se nd u	se na ng	Kegiata n Berfok us	Bero lahra ga	Isti rah at	
1	6	1	0	0	0	0	0	0	0	1	0	0	klas ik
2	5	1	1	0	0	0	0	0	0	1	0	0	klas ik
3	4	1	1	0	0	0	0	1	0	1	0	0	klas ik
4	6	1	1	0	0	0	0	0	0	0	0	1	klas ik
5	4	0	1	0	0	0	0	0	0	1	0	0	klas ik
6	6	0	1	0	0	0	0	0	0	0	0	1	Mu rott al
7	5	0	1	0	0	0	0	0	1	0	0	1	Mu rott al
8	5	0	1	0	0	0	0	1	0	0	0	1	Mu rott al
9	4	0	1	0	0	0	0	1	1	0	0	1	Mu rott al
10	5	0	1	0	0	1	0	0	0	0	0	1	Mu rott al
11	4	0	0	0	0	1	0	0	0	0	1	0	roc k
12	5	0	0	0	0	1	1	0	0	0	1	0	roc k
13	6	0	0	1	0	1	0	0	0	0	1	0	roc k
14	4	0	0	0	0	1	0	0	0	0	0	1	roc k
15	6	0	0	1	0	1	1	0	0	0	1	0	roc k
16	5	0	0	0	0	0	0	1	0	0	0	1	po p

D A T A	Juml ah Kasus	FITUR											KEL AS
		kons entr asi	te na ng	sem ang at	c er ia	be ba s	bert ena ga	se nd u	se na ng	Kegiata n Berfok us	Bero lahra ga	Isti rah at	
17	5	0	0	0	0	0	0	1	1	0	0	1	po p
18	5	0	0	0	1	0	0	0	1	0	0	1	po p
19	4	0	1	0	0	0	0	1	0	0	0	1	po p
20	6	0	1	0	0	0	0	1	0	0	0	1	po p

B.2 Data Awal (D2)

Penjabaran dari Data awal/D1 adalah Data awal tanpa jumlah kasus atau *dataset* 2 (D2).

D at a	FITUR											KEL AS
	konse ntrasi	ten an g	sem anga t	ce ri a	be ba s	bert enag a	se nd u	sen an g	Kegiatan Berfokus	Berol ahrag a	Istir aha t	
1	1	0	0	0	0	0	0	0	1	0	0	klasi k
2	1	0	0	0	0	0	0	0	1	0	0	klasi k
3	1	0	0	0	0	0	0	0	1	0	0	klasi k
4	1	0	0	0	0	0	0	0	1	0	0	klasi k
5	1	0	0	0	0	0	0	0	1	0	0	klasi k
6	1	0	0	0	0	0	0	0	1	0	0	klasi k
7	1	1	0	0	0	0	0	0	1	0	0	klasi k
8	1	1	0	0	0	0	0	0	1	0	0	klasi k
9	1	1	0	0	0	0	0	0	1	0	0	klasi k
10	1	1	0	0	0	0	0	0	1	0	0	klasi k
11	1	1	0	0	0	0	0	0	1	0	0	klasi k
12	1	1	0	0	0	0	1	0	1	0	0	klasi k
13	1	1	0	0	0	0	1	0	1	0	0	klasi k
14	1	1	0	0	0	0	1	0	1	0	0	klasi k
15	1	1	0	0	0	0	1	0	1	0	0	klasi k
16	1	1	0	0	0	0	0	0	0	0	1	klasi k
17	1	1	0	0	0	0	0	0	0	0	1	klasi k
18	1	1	0	0	0	0	0	0	0	0	1	klasi k
19	1	1	0	0	0	0	0	0	0	0	1	klasi k
20	1	1	0	0	0	0	0	0	0	0	1	klasi k

D at a	FITUR											KEL AS
	konse ntrasi	ten an g	sem anga t	ce ri a	be ba s	bert enag a	se nd u	sen an g	Kegiatan Berfokus	Berol ahrag a	Istir aha t	
2 1	1	1	0	0	0	0	0	0	0	0	1	klasi k
2 2	0	1	0	0	0	0	0	0	1	0	0	klasi k
2 3	0	1	0	0	0	0	0	0	1	0	0	klasi k
2 4	0	1	0	0	0	0	0	0	1	0	0	klasi k
2 5	0	1	0	0	0	0	0	0	1	0	0	klasi k
2 6	0	1	0	0	0	0	0	0	0	0	1	Mur otta l
2 7	0	1	0	0	0	0	0	0	0	0	1	Mur otta l
2 8	0	1	0	0	0	0	0	0	0	0	1	Mur otta l
2 9	0	1	0	0	0	0	0	0	0	0	1	Mur otta l
3 0	0	1	0	0	0	0	0	0	0	0	1	Mur otta l
3 1	0	1	0	0	0	0	0	0	0	0	1	Mur otta l
3 2	0	1	0	0	0	0	0	1	0	0	1	Mur otta l
3 3	0	1	0	0	0	0	0	1	0	0	1	Mur otta l
3 4	0	1	0	0	0	0	0	1	0	0	1	Mur otta l
3 5	0	1	0	0	0	0	0	1	0	0	1	Mur otta l
3 6	0	1	0	0	0	0	0	1	0	0	1	Mur otta l
3 7	0	1	0	0	0	0	1	0	0	0	1	Mur otta

D at a	FITUR											KEL AS
	konse ntrasi	ten an g	sem anga t	ce ri a	be ba s	bert enag a	se nd u	sen an g	Kegiatan Berfokus	Berol ahrag a	Istir aha t	
												I
3 8	0	1	0	0	0	0	1	0	0	0	1	Mur otta I
3 9	0	1	0	0	0	0	1	0	0	0	1	Mur otta I
4 0	0	1	0	0	0	0	1	0	0	0	1	Mur otta I
4 1	0	1	0	0	0	0	1	0	0	0	1	Mur otta I
4 2	0	1	0	0	0	0	1	1	0	0	1	Mur otta I
4 3	0	1	0	0	0	0	1	1	0	0	1	Mur otta I
4 4	0	1	0	0	0	0	1	1	0	0	1	Mur otta I
4 5	0	1	0	0	0	0	1	1	0	0	1	Mur otta I
4 6	0	1	0	0	1	0	0	0	0	0	1	Mur otta I
4 7	0	1	0	0	1	0	0	0	0	0	1	Mur otta I
4 8	0	1	0	0	1	0	0	0	0	0	1	Mur otta I
4 9	0	1	0	0	1	0	0	0	0	0	1	Mur otta I
5 0	0	1	0	0	1	0	0	0	0	0	1	Mur otta I
5 1	0	0	0	0	1	0	0	0	0	1	0	rock
5 2	0	0	0	0	1	0	0	0	0	1	0	rock

D a t a	FITUR											KEL AS
	konse ntrasi	ten an g	sem anga t	ce ri a	be ba s	bert enag a	se nd u	sen an g	Kegiatan Berfokus	Berol ahrag a	Istir aha t	
53	0	0	0	0	1	0	0	0	0	1	0	rock
54	0	0	0	0	1	0	0	0	0	1	0	rock
55	0	0	0	0	1	1	0	0	0	1	0	rock
56	0	0	0	0	1	1	0	0	0	1	0	rock
57	0	0	0	0	1	1	0	0	0	1	0	rock
58	0	0	0	0	1	1	0	0	0	1	0	rock
59	0	0	0	0	1	1	0	0	0	1	0	rock
60	0	0	1	0	1	0	0	0	0	1	0	rock
61	0	0	1	0	1	0	0	0	0	1	0	rock
62	0	0	1	0	1	0	0	0	0	1	0	rock
63	0	0	1	0	1	0	0	0	0	1	0	rock
64	0	0	1	0	1	0	0	0	0	1	0	rock
65	0	0	1	0	1	0	0	0	0	1	0	rock
66	0	0	0	0	1	0	0	0	0	0	1	rock
67	0	0	0	0	1	0	0	0	0	0	1	rock
68	0	0	0	0	1	0	0	0	0	0	1	rock
69	0	0	0	0	1	0	0	0	0	0	1	rock
70	0	0	1	0	1	1	0	0	0	1	0	rock
71	0	0	1	0	1	1	0	0	0	1	0	rock
72	0	0	1	0	1	1	0	0	0	1	0	rock
73	0	0	1	0	1	1	0	0	0	1	0	rock
74	0	0	1	0	1	1	0	0	0	1	0	rock

D at a	FITUR											KEL AS
	konse ntrasi	ten an g	sem anga t	ce ri a	be ba s	bert enag a	se nd u	sen an g	Kegiatan Berfokus	Berol ahrag a	Istir aha t	
7 5	0	0	1	0	1	1	0	0	0	1	0	rock
7 6	0	0	0	0	0	0	1	0	0	0	1	pop
7 7	0	0	0	0	0	0	1	0	0	0	1	pop
7 8	0	0	0	0	0	0	1	0	0	0	1	pop
7 9	0	0	0	0	0	0	1	0	0	0	1	pop
8 0	0	0	0	0	0	0	1	0	0	0	1	pop
8 1	0	0	0	0	0	0	1	1	0	0	1	pop
8 2	0	0	0	0	0	0	1	1	0	0	1	pop
8 3	0	0	0	0	0	0	1	1	0	0	1	pop
8 4	0	0	0	0	0	0	1	1	0	0	1	pop
8 5	0	0	0	0	0	0	1	1	0	0	1	pop
8 6	0	0	0	1	0	0	0	1	0	0	1	pop
8 7	0	0	0	1	0	0	0	1	0	0	1	pop
8 8	0	0	0	1	0	0	0	1	0	0	1	pop
8 9	0	0	0	1	0	0	0	1	0	0	1	pop
9 0	0	0	0	1	0	0	0	1	0	0	1	pop
9 1	0	1	0	0	0	0	1	0	0	0	1	pop
9 2	0	1	0	0	0	0	1	0	0	0	1	pop
9 3	0	1	0	0	0	0	1	0	0	0	1	pop
9 4	0	1	0	0	0	0	1	0	0	0	1	pop
9 5	0	1	0	0	0	0	1	0	0	0	1	pop
9 6	0	1	0	0	0	0	1	0	0	0	1	pop

D a t a	FITUR											KEL AS
	konse ntrasi	ten an g	sem ang at	ce ri a	be ba s	bert enag a	se nd u	sen an g	Kegiatan Berfokus	Berol ahrag a	Istir aha t	
9 7	0	1	0	0	0	0	0	1	1	0	0	pop
9 8	0	1	0	0	0	0	0	1	1	0	0	pop
9 9	0	1	0	0	0	0	0	1	1	0	0	pop
1 0 0	0	1	0	0	0	0	0	1	1	0	0	pop

B.3 Data Lanjutan (D3)

Data lanjutan atau *dataset* 3 (D3) adalah perkembangan dari data awal dikarenakan pengumpulan data dilanjutkan. D3 memiliki jumlah kasus dari tiap data yang memiliki kesamaan.

D A T A	Juml ah Kasus	FITUR											KEL AS
		kons entrasi	te na ng	sem ang at	c er ia	be ba s	bert ena ga	se nd u	se na ng	Kegiata n Berfok us	Bero lahra ga	Isti rah at	
1	6	1	0	0	0	0	0	0	0	1	0	0	klas ik
2	10	1	1	0	0	0	0	0	0	1	0	0	klas ik
3	4	1	1	0	0	0	0	1	0	1	0	0	klas ik
4	6	1	1	0	0	0	0	0	0	0	0	1	klas ik
5	4	0	1	0	0	0	0	0	0	1	0	0	klas ik
6	3	1	1	0	0	0	0	1	0	1	0	0	klas ik
7	3	0	0	0	0	0	0	1	0	1	0	1	klas ik
8	4	1	1	0	0	0	0	0	1	1	0	0	klas ik
9	10	0	1	0	0	0	0	0	0	0	0	1	Mu rott al
10	6	0	1	0	0	0	0	0	1	0	0	1	Mu rott al
11	5	0	1	0	0	0	0	1	0	0	0	1	Mu

D A T A	Juml ah Kasus	FITUR											KEL AS
		kons entr asi	te na ng	sem ang at	c er ia	be ba s	bert ena ga	se nd u	se na ng	Kegiata n Berfok us	Bero lahra ga	Isti rah at	
													rott al
12	4	0	1	0	0	0	0	1	1	0	0	1	Mu rott al
13	5	0	1	0	0	1	0	0	0	0	0	1	Mu rott al
14	4	1	1	0	0	0	0	1	0	1	0	1	Mu rott al
15	6	1	0	0	0	0	0	1	0	1	0	0	Mu rott al
16	8	0	0	0	0	1	0	0	0	0	1	0	roc k
17	5	0	0	0	0	1	1	0	0	0	1	0	roc k
18	7	0	0	1	0	1	0	0	0	0	1	0	roc k
19	4	0	0	0	0	1	0	0	0	0	0	1	roc k
20	6	0	0	1	0	1	1	0	0	0	1	0	roc k
21	5	0	0	1	1	1	1	0	0	1	1	0	roc k
22	3	0	0	1	1	1	1	0	1	0	1	0	roc k
23	2	1	0	1	0	0	0	0	0	0	1	0	roc k
24	7	0	0	0	0	0	0	1	0	0	0	1	po p
25	6	0	0	0	0	0	0	1	1	0	0	1	po p
26	7	0	0	0	1	0	0	0	1	0	0	1	po p
27	4	0	1	0	0	0	0	1	0	0	0	1	po p
28	8	0	1	0	0	0	0	1	0	0	0	1	po p
29	2	0	0	0	1	0	0	0	1	0	1	1	po p
30	3	1	0	1	1	0	0	0	0	0	1	0	po p

D A T A	Juml ah Kasus	FITUR											KEL AS
		kons entr asi	te na ng	sem ang at	c er ia	be ba s	bert ena ga	se nd u	se na ng	Kegiata n Berfok us	Bero lahra ga	Isti rah at	
31	3	0	1	0	0	0	0	1	0	1	0	1	po p

B.4 Data Lanjutan (D4)

Penjabaran dari Data Lanjutan/D3 adalah Data awal tanpa jumlah kasus atau dataset 4 (D4).

D at a	FITUR											KEL AS
	konse ntrasi	ten an g	sem anga t	ce ri a	be ba s	bert enag a	se nd u	sen an g	Kegiatan Berfokus	Berol ahrag a	Istir aha t	
1	1	0	0	0	0	0	0	0	1	0	0	klasi k
2	1	0	0	0	0	0	0	0	1	0	0	klasi k
3	1	0	0	0	0	0	0	0	1	0	0	klasi k
4	1	0	0	0	0	0	0	0	1	0	0	klasi k
5	1	0	0	0	0	0	0	0	1	0	0	klasi k
6	1	0	0	0	0	0	0	0	1	0	0	klasi k
7	1	1	0	0	0	0	0	0	1	0	0	klasi k
8	1	1	0	0	0	0	0	0	1	0	0	klasi k
9	1	1	0	0	0	0	0	0	1	0	0	klasi k
10	1	1	0	0	0	0	0	0	1	0	0	klasi k
11	1	1	0	0	0	0	0	0	1	0	0	klasi k
12	1	1	0	0	0	0	0	0	1	0	0	klasi k
13	1	1	0	0	0	0	0	0	1	0	0	klasi k
14	1	1	0	0	0	0	0	0	1	0	0	klasi k
15	1	1	0	0	0	0	0	0	1	0	0	klasi k
16	1	1	0	0	0	0	0	0	1	0	0	klasi k

D a t a	FITUR											KEL AS
	konse ntrasi	ten an g	sem anga t	ce ri a	be ba s	bert enag a	se nd u	sen an g	Kegiatan Berfokus	Berol ahrag a	Istir aha t	
17	1	1	0	0	0	0	1	0	1	0	0	klasi k
18	1	1	0	0	0	0	1	0	1	0	0	klasi k
19	1	1	0	0	0	0	1	0	1	0	0	klasi k
20	1	1	0	0	0	0	1	0	1	0	0	klasi k
21	1	1	0	0	0	0	0	0	0	0	1	klasi k
22	1	1	0	0	0	0	0	0	0	0	1	klasi k
23	1	1	0	0	0	0	0	0	0	0	1	klasi k
24	1	1	0	0	0	0	0	0	0	0	1	klasi k
25	1	1	0	0	0	0	0	0	0	0	1	klasi k
26	1	1	0	0	0	0	0	0	0	0	1	klasi k
27	0	1	0	0	0	0	0	0	1	0	0	klasi k
28	0	1	0	0	0	0	0	0	1	0	0	klasi k
29	0	1	0	0	0	0	0	0	1	0	0	klasi k
30	0	1	0	0	0	0	0	0	1	0	0	klasi k
31	1	1	0	0	0	0	1	0	1	0	0	klasi k
32	1	1	0	0	0	0	1	0	1	0	0	klasi k
33	1	1	0	0	0	0	1	0	1	0	0	klasi k
34	0	0	0	0	0	0	1	0	1	0	1	klasi k
35	0	0	0	0	0	0	1	0	1	0	1	klasi k
36	0	0	0	0	0	0	1	0	1	0	1	klasi k
37	1	1	0	0	0	0	0	1	1	0	0	klasi k
38	1	1	0	0	0	0	0	1	1	0	0	klasi k

D at a	FITUR											KEL AS
	konse ntrasi	ten an g	sem anga t	ce ri a	be ba s	bert enag a	se nd u	sen an g	Kegiatan Berfokus	Berol ahrag a	Istir aha t	
3 9	1	1	0	0	0	0	0	1	1	0	0	klasi k
4 0	1	1	0	0	0	0	0	1	1	0	0	klasi k
4 1	0	1	0	0	0	0	0	0	0	0	1	Mur otta l
4 2	0	1	0	0	0	0	0	0	0	0	1	Mur otta l
4 3	0	1	0	0	0	0	0	0	0	0	1	Mur otta l
4 4	0	1	0	0	0	0	0	0	0	0	1	Mur otta l
4 5	0	1	0	0	0	0	0	0	0	0	1	Mur otta l
4 6	0	1	0	0	0	0	0	0	0	0	1	Mur otta l
4 7	0	1	0	0	0	0	0	0	0	0	1	Mur otta l
4 8	0	1	0	0	0	0	0	0	0	0	1	Mur otta l
4 9	0	1	0	0	0	0	0	0	0	0	1	Mur otta l
5 0	0	1	0	0	0	0	0	0	0	0	1	Mur otta l
5 1	0	1	0	0	0	0	0	1	0	0	1	Mur otta l
5 2	0	1	0	0	0	0	0	1	0	0	1	Mur otta l
5 3	0	1	0	0	0	0	0	1	0	0	1	Mur otta l
5 4	0	1	0	0	0	0	0	1	0	0	1	Mur otta

D at a	FITUR											KEL AS
	konse ntrasi	ten an g	sem anga t	ce ri a	be ba s	bert enag a	se nd u	sen an g	Kegiatan Berfokus	Berol ahrag a	Istir aha t	
												I
5 5	0	1	0	0	0	0	0	1	0	0	1	Mur otta I
5 6	0	1	0	0	0	0	0	1	0	0	1	Mur otta I
5 7	0	1	0	0	0	0	1	0	0	0	1	Mur otta I
5 8	0	1	0	0	0	0	1	0	0	0	1	Mur otta I
5 9	0	1	0	0	0	0	1	0	0	0	1	Mur otta I
6 0	0	1	0	0	0	0	1	0	0	0	1	Mur otta I
6 1	0	1	0	0	0	0	1	0	0	0	1	Mur otta I
6 2	0	1	0	0	0	0	1	1	0	0	1	Mur otta I
6 3	0	1	0	0	0	0	1	1	0	0	1	Mur otta I
6 4	0	1	0	0	0	0	1	1	0	0	1	Mur otta I
6 5	0	1	0	0	0	0	1	1	0	0	1	Mur otta I
6 6	0	1	0	0	1	0	0	0	0	0	1	Mur otta I
6 7	0	1	0	0	1	0	0	0	0	0	1	Mur otta I
6 8	0	1	0	0	1	0	0	0	0	0	1	Mur otta I
6 9	0	1	0	0	1	0	0	0	0	0	1	Mur otta

D at a	FITUR											KEL AS
	konse ntrasi	ten an g	sem anga t	ce ri a	be ba s	bert enag a	se nd u	sen an g	Kegiatan Berfokus	Berol ahrag a	Istir aha t	
												I
7 0	0	1	0	0	1	0	0	0	0	0	1	Mur otta I
7 1	1	1	0	0	0	0	1	0	1	0	1	Mur otta I
7 2	1	1	0	0	0	0	1	0	1	0	1	Mur otta I
7 3	1	1	0	0	0	0	1	0	1	0	1	Mur otta I
7 4	1	1	0	0	0	0	1	0	1	0	1	Mur otta I
7 5	1	0	0	0	0	0	1	0	1	0	0	Mur otta I
7 6	1	0	0	0	0	0	1	0	1	0	0	Mur otta I
7 7	1	0	0	0	0	0	1	0	1	0	0	Mur otta I
7 8	1	0	0	0	0	0	1	0	1	0	0	Mur otta I
7 9	1	0	0	0	0	0	1	0	1	0	0	Mur otta I
8 0	1	0	0	0	0	0	1	0	1	0	0	Mur otta I
8 1	0	0	0	0	1	0	0	0	0	1	0	rock
8 2	0	0	0	0	1	0	0	0	0	1	0	rock
8 3	0	0	0	0	1	0	0	0	0	1	0	rock
8 4	0	0	0	0	1	0	0	0	0	1	0	rock
8 5	0	0	0	0	1	1	0	0	0	1	0	rock

D at a	FITUR											KEL AS
	konse ntrasi	ten an g	sem anga t	ce ri a	be ba s	bert enag a	se nd u	sen an g	Kegiatan Berfokus	Berol ahrag a	Istir aha t	
8 6	0	0	0	0	1	1	0	0	0	1	0	rock
8 7	0	0	0	0	1	1	0	0	0	1	0	rock
8 8	0	0	0	0	1	1	0	0	0	1	0	rock
8 9	0	0	0	0	1	1	0	0	0	1	0	rock
9 0	0	0	1	0	1	0	0	0	0	1	0	rock
9 1	0	0	1	0	1	0	0	0	0	1	0	rock
9 2	0	0	1	0	1	0	0	0	0	1	0	rock
9 3	0	0	1	0	1	0	0	0	0	1	0	rock
9 4	0	0	1	0	1	0	0	0	0	1	0	rock
9 5	0	0	1	0	1	0	0	0	0	1	0	rock
9 6	0	0	1	0	1	0	0	0	0	1	0	rock
9 7	0	0	0	0	1	0	0	0	0	0	1	rock
9 8	0	0	0	0	1	0	0	0	0	0	1	rock
9 9	0	0	0	0	1	0	0	0	0	0	1	rock
1 0 0	0	0	0	0	1	0	0	0	0	0	1	rock
1 0 1	0	0	1	0	1	1	0	0	0	1	0	rock
1 0 2	0	0	1	0	1	1	0	0	0	1	0	rock
1 0 3	0	0	1	0	1	1	0	0	0	1	0	rock
1 0 4	0	0	1	0	1	1	0	0	0	1	0	rock
1 0	0	0	1	0	1	1	0	0	0	1	0	rock

D at a	FITUR											KEL AS
	konse ntrasi	ten an g	sem anga t	ce ri a	be ba s	bert enag a	se nd u	sen an g	Kegiatan Berfokus	Berol ahrag a	Istir aha t	
5												
1 0 6	0	0	1	0	1	1	0	0	0	1	0	rock
1 0 7	0	0	1	0	1	1	0	0	0	1	0	rock
1 0 8	0	0	1	0	1	1	0	0	0	1	0	rock
1 0 9	0	0	1	0	1	1	0	0	0	1	0	rock
1 1 0	0	0	1	0	1	1	0	0	0	1	0	rock
1 1 1	0	0	1	1	1	1	0	0	1	1	0	rock
1 1 2	0	0	1	1	1	1	0	0	1	1	0	rock
1 1 3	0	0	1	1	1	1	0	0	1	1	0	rock
1 1 4	0	0	1	1	1	1	0	0	1	1	0	rock
1 1 5	0	0	1	1	1	1	0	0	1	1	0	rock
1 1 6	0	0	1	1	1	1	0	1	0	1	0	rock
1 1 7	0	0	1	1	1	1	0	1	0	1	0	rock
1 1 8	0	0	1	1	1	1	0	1	0	1	0	rock
1 1 9	1	0	1	0	0	0	0	0	0	1	0	rock
1 2	1	0	1	0	0	0	0	0	0	1	0	rock

D at a	FITUR											KEL AS
	konse ntrasi	ten an g	sem anga t	ce ri a	be ba s	bert enag a	se nd u	sen an g	Kegiatan Berfokus	Berol ahrag a	Istir aha t	
0												
1 2 1	0	0	0	0	0	0	1	0	0	0	1	pop
1 2 2	0	0	0	0	0	0	1	0	0	0	1	pop
1 2 3	0	0	0	0	0	0	1	0	0	0	1	pop
1 2 4	0	0	0	0	0	0	1	0	0	0	1	pop
1 2 5	0	0	0	0	0	0	1	0	0	0	1	pop
1 2 6	0	0	0	0	0	0	1	1	0	0	1	pop
1 2 7	0	0	0	0	0	0	1	1	0	0	1	pop
1 2 8	0	0	0	0	0	0	1	1	0	0	1	pop
1 2 9	0	0	0	0	0	0	1	1	0	0	1	pop
1 3 0	0	0	0	0	0	0	1	1	0	0	1	pop
1 3 1	0	0	0	1	0	0	0	1	0	0	1	pop
1 3 2	0	0	0	1	0	0	0	1	0	0	1	pop
1 3 3	0	0	0	1	0	0	0	1	0	0	1	pop
1 3 4	0	0	0	1	0	0	0	1	0	0	1	pop
1 3	0	0	0	1	0	0	0	1	0	0	1	pop

D a t a	FITUR											KEL AS
	konse ntrasi	ten an g	sem anga t	ce ri a	be ba s	bert enag a	se nd u	sen an g	Kegiatan Berfokus	Berol ahrag a	Istir aha t	
5												
1 3 6	0	0	0	1	0	0	0	1	0	0	1	pop
1 3 7	0	0	0	1	0	0	0	1	0	0	1	pop
1 3 8	0	0	0	1	0	0	0	1	0	0	1	pop
1 3 9	0	0	0	1	0	0	0	1	0	0	1	pop
1 4 0	0	0	0	1	0	0	0	1	0	0	1	pop
1 4 1	0	1	0	0	0	0	1	0	0	0	1	pop
1 4 2	0	1	0	0	0	0	1	0	0	0	1	pop
1 4 3	0	1	0	0	0	0	1	0	0	0	1	pop
1 4 4	0	1	0	0	0	0	1	0	0	0	1	pop
1 4 5	0	1	0	0	0	0	1	0	0	0	1	pop
1 4 6	0	1	0	0	0	0	1	0	0	0	1	pop
1 4 7	0	1	0	0	0	0	0	1	1	0	0	pop
1 4 8	0	1	0	0	0	0	0	1	1	0	0	pop
1 4 9	0	1	0	0	0	0	0	1	1	0	0	pop
1 5	0	1	0	0	0	0	0	1	1	0	0	pop

D a t a	FITUR											KEL AS
	konse ntrasi	ten an g	sem anga t	ce ri a	be ba s	bert enag a	se nd u	sen an g	Kegiatan Berfokus	Berol ahrag a	Istir aha t	
0												
1 5 1	0	0	0	1	0	0	0	1	0	1	1	pop
1 5 2	0	0	0	1	0	0	0	1	0	1	1	pop
1 5 3	0	0	0	1	0	0	0	1	0	1	1	pop
1 5 4	0	0	0	1	0	0	0	1	0	1	1	pop
1 5 5	1	0	1	1	0	0	0	0	0	1	0	pop
1 5 6	1	0	1	1	0	0	0	0	0	1	0	pop
1 5 7	1	0	1	1	0	0	0	0	0	1	0	pop
1 5 8	0	1	0	0	0	0	1	0	1	0	1	pop
1 5 9	0	1	0	0	0	0	1	0	1	0	1	pop
1 6 0	0	1	0	0	0	0	1	0	1	0	1	pop